

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日                      2 0 0 3 年   8 月 1 1 日  
Date of Application:

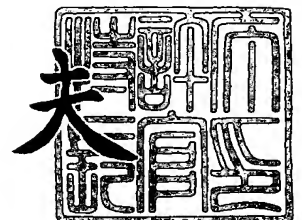
出 願 番 号                      特 願 2 0 0 3 - 2 9 1 0 4 5  
Application Number:  
[ST. 10/C]:                      [ J P 2 0 0 3 - 2 9 1 0 4 5 ]

出      願      人                      株 式 会 社 リ コ ー  
Applicant(s):

2 0 0 3 年   9 月   1 日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

今 井 康 夫



【書類名】 特許願  
【整理番号】 0305842  
【提出日】 平成15年 8月11日  
【あて先】 特許庁長官 今井 康夫 殿  
【国際特許分類】 G03G 21/00 370  
【発明者】  
    【住所又は居所】 東京都大田区中馬込 1 丁目 3 番 6 号 株式会社リコー内  
    【氏名】 加藤 良一  
【発明者】  
    【住所又は居所】 東京都大田区中馬込 1 丁目 3 番 6 号 株式会社リコー内  
    【氏名】 阿部 良彦  
【発明者】  
    【住所又は居所】 東京都大田区中馬込 1 丁目 3 番 6 号 株式会社リコー内  
    【氏名】 杉下 悟  
【発明者】  
    【住所又は居所】 東京都大田区中馬込 1 丁目 3 番 6 号 株式会社リコー内  
    【氏名】 佐々木 勝彦  
【特許出願人】  
    【識別番号】 000006747  
    【氏名又は名称】 株式会社リコー  
【代理人】  
    【識別番号】 100070150  
    【弁理士】  
    【氏名又は名称】 伊東 忠彦  
【先の出願に基づく優先権主張】  
    【出願番号】 特願2002-247448  
    【出願日】 平成14年 8月27日  
【先の出願に基づく優先権主張】  
    【出願番号】 特願2002-249629  
    【出願日】 平成14年 8月28日  
【先の出願に基づく優先権主張】  
    【出願番号】 特願2002-256761  
    【出願日】 平成14年 9月 2日  
【手数料の表示】  
    【予納台帳番号】 002989  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 9911477

**【書類名】 特許請求の範囲****【請求項 1】**

画像形成処理で使用するハードウェア資源と、画像形成に係る処理をプログラムに基づき実行するプロセスとを有する画像形成装置において、

前記プロセスの処理の実行を制限させた状態であるオフライン状態に前記プロセスを移行させるオフライン手段と、

前記オフライン状態に移行した前記プロセスが使用していた記憶領域を解放する記憶領域解放手段と、

前記記憶領域解放手段が解放した記憶領域を、データを展開する記憶領域として使用するデータ展開手段と

を有することを特徴とする画像形成装置。

**【請求項 2】**

前記データ展開手段は、前記記憶領域に前記データを展開する際に、前記オフライン手段により、前記プロセスをオフライン状態に移行させることを特徴とする請求項 1 に記載の画像形成装置。

**【請求項 3】**

前記データ展開手段は、前記オフライン手段が前記プロセスをオフライン状態に移行させると、前記記憶領域解放手段により、オフライン状態に移行した前記プロセスが使用していた記憶領域を解放させることを特徴とする請求項 2 に記載の画像形成装置。

**【請求項 4】**

前記オフライン手段は、前記プロセスに対し、前記プロセスをオフライン状態に移行させるオフライン移行要求を通知して、前記プロセスをオフライン状態に移行させることを特徴とする請求項 1 に記載の画像形成装置。

**【請求項 5】**

前記オフライン手段は、前記プロセスからの前記オフライン移行要求に対する応答に基づき、前記プロセスがオフライン状態に移行したかどうかを前記データ展開手段に通知することを特徴とする請求項 4 に記載の画像形成装置。

**【請求項 6】**

前記オフライン手段は、前記オフライン移行要求を通知した全てのプロセスがオフライン状態に移行すると、前記データ展開手段に対し、前記プロセスがオフライン状態に移行したことを通知することを特徴とする請求項 5 に記載の画像形成装置。

**【請求項 7】**

前記オフライン手段は、前記オフライン移行要求を通知した前記プロセスから、オフライン状態に移行できないことを通知されると、前記データ展開手段に対し、前記プロセスがオフライン状態に移行しなかったことを通知することを特徴とする請求項 5 に記載の画像形成装置。

**【請求項 8】**

前記オフライン手段は、前記オフライン移行要求を通知した前記プロセスのうち、1つ以上のプロセスから、オフライン状態に移行できないことを通知されても、前記オフライン移行要求を通知した全てのプロセスからの応答を待ったのちに、前記データ展開手段に対し、前記プロセスがオフライン状態に移行しなかったことを通知することを特徴とする請求項 5 に記載の画像形成装置。

**【請求項 9】**

前記オフライン手段は、前記オフライン移行要求を通知した前記プロセスのうち、1つのプロセスからオフライン状態に移行できないことを通知されると、前記オフライン移行要求を通知した他のプロセスからの応答を待たずに、前記データ展開手段に対し、前記プロセスがオフライン状態に移行しなかったことを通知することを特徴とする請求項 5 に記載の画像形成装置。

**【請求項 10】**

前記オフライン手段は、前記プロセスに前記オフライン移行要求を通知してからの時間を

計測し、

前記オフライン移行要求に対する前記プロセスの応答が、所定の時間が経過しても通知されない場合、前記データ展開手段に対し、前記プロセスがオフライン状態に移行したことを通知することを特徴とする請求項 4 に記載の画像形成装置。

【請求項 1 1】

オフライン状態にされた前記プロセスは、前記オフライン手段がオフライン状態を解除することにより、制限されていた処理の実行が可能となることを特徴とする請求項 1 に記載の画像形成装置。

【請求項 1 2】

前記オフライン状態に移行したプロセスが実行を制限される処理は、他のプロセスからの要求により実行される処理であることを特徴とする請求項 1 に記載の画像形成装置。

【請求項 1 3】

前記オフライン状態に移行したプロセスは、他のプロセスからの要求を保持しておくことを特徴とする請求項 1 2 に記載の画像形成装置。

【請求項 1 4】

前記オフライン状態に移行したプロセスを消滅させるプロセス消滅手段をさらに有することを特徴とする請求項 1 に記載の画像形成装置。

【請求項 1 5】

前記プロセス消滅手段は、各プロセスに予め定められた順番で前記プロセスを消滅することを特徴とする請求項 1 4 に記載の画像形成装置。

【請求項 1 6】

前記順番は、各プロセスに予め定められた優先度に応じて定まることを特徴とする請求項 1 5 に記載の画像形成装置。

【請求項 1 7】

前記順番は、各プロセスに予め割り当てられた記憶領域のサイズに応じて定まることを特徴とする請求項 1 5 に記載の画像形成装置。

【請求項 1 8】

前記順番は、前記プロセスに割り当てられた記憶領域の位置に応じて定まることを特徴とする請求項 1 5 に記載の画像形成装置。

【請求項 1 9】

前記記憶領域解放手段は、前記プロセス消滅手段がプロセスを消滅させたことにより不要となった記憶領域を解放することを特徴とする請求項 1 4 に記載の画像形成装置。

【請求項 2 0】

前記記憶領域解放手段は、展開される前記データのサイズに応じて、前記記憶領域を解放することを特徴とする請求項 1 9 に記載の画像形成装置。

【請求項 2 1】

前記記憶領域解放手段は、前記記憶領域の解放を行うと、前記データ展開手段に対し、前記記憶領域の解放が完了したことを通知することを特徴とする請求項 1 に記載の画像形成装置。

【請求項 2 2】

前記データ展開手段が展開するデータは、前記プログラムを更新する更新プログラムであり、前記データ展開手段は、前記更新プログラムを通信手段を用いて取得することを特徴とする請求項 1 に記載の画像形成装置。

【請求項 2 3】

前記データ展開手段からのプログラム更新開始要求により、前記プログラムを更新するプログラム更新手段をさらに有することを特徴とする請求項 2 2 に記載の画像形成装置。

【請求項 2 4】

前記画像形成装置を操作するための入力手段を有し、

前記プログラム更新手段は、前記プログラムを更新する際に、前記入力手段を無効化することを特徴とする請求項 2 3 に画像形成装置。

**【請求項 25】**

前記プログラム更新手段は、前記プログラムの更新が終了すると、前記画像形成装置を再起動することを特徴とする請求項 23 に記載の画像形成装置。

**【請求項 26】**

前記プログラム更新手段は、前記画像形成装置と通信可能な機器に対し、前記プログラムの更新状況を通知することを特徴とする請求項 23 に記載の画像形成装置。

**【請求項 27】**

前記プログラム更新手段は、前記プログラムの更新状況を、オフライン状態に移行したプロセスを用いて通知することを特徴とする請求項 26 に記載の画像形成装置。

**【請求項 28】**

前記記憶領域は、前記プログラムの実行および前記ハードウェア資源を管理するオペレーティングシステムの管理外にあることを特徴とする請求項 1 に記載の画像形成装置。

**【請求項 29】**

画像形成処理で使用されるハードウェア資源と、画像形成に係る処理をプログラムに基づき実行するプロセスと、前記プログラムの実行および前記ハードウェア資源を管理するオペレーティングシステムとを有する画像形成装置の記憶領域確保方法であって、

前記プロセスに対し、処理の実行を制限させるオフライン段階と、

前記オペレーティングシステムの管理外にあり、前記オフライン段階で処理の実行が制限された前記プロセスが使用していた記憶領域を解放する記憶領域解放段階と

前記記憶領域解放手段が解放した記憶領域に、データを展開するデータ展開段階とを有することを特徴とする記憶領域確保方法。

**【請求項 30】**

前記記憶領域解放段階で、前記プロセスがオフライン状態に移行した後、前記プロセスを消滅させるプロセス消滅段階をさらに有することを特徴とする請求項 29 に記載の記憶領域確保方法。

**【請求項 31】**

前記データ展開段階で展開されるデータは、前記プログラムを更新する更新プログラムであることを特徴とする請求項 29 に記載の記憶領域確保方法。

【書類名】明細書

【発明の名称】画像形成装置、記憶領域確保方法

【技術分野】

【0001】

本発明は、画像形成装置に関し、特にメモリ資源を有効に利用する画像形成装置、記憶領域確保方法に関する。

【背景技術】

【0002】

近年、プリンタ、コピー、ファクシミリおよびスキャナなどの各装置の機能を1つの筐体内に収納した画像形成装置（以下、融合機という）が知られるようになった。この融合機は、1つの筐体内に表示部、印刷部および撮像部などを設けると共に、プリンタ、コピー、ファクシミリおよびスキャナにそれぞれ対応する4種類のアプリケーションを設け、そのアプリケーションを切り替えることより、プリンタ、コピー、ファクシミリおよびスキャナとして動作させるものである。

【0003】

このように、融合機は、プログラムに基づき処理を実行する種々のアプリケーションにより動作しているため、プログラムを更新することにより、新たな機能の追加などを行うことができる。

【0004】

このプログラムの更新処理は、パソコンなどでよく見られるように、インターネットなどのネットワークを介し、更新プログラムが格納されたサーバに接続し更新プログラムを取得（以下、ダウンロードと記す）することにより行われることがある。

【0005】

この場合、サーバからダウンロードした更新プログラムをメモリに展開し、フラッシュメモリなどのEEPROM(Electrically Erasable and Programmable ROM)に更新プログラムを書き込むことで、プログラムを更新することができる。

【0006】

このように、サーバからダウンロードした更新プログラムはメモリに展開されるため、メモリには更新プログラムを展開するための記憶領域（以下、メモリ領域と記す）が必要となっている。

【0007】

このメモリ領域は、アプリケーションなどが処理を実行する際に確保したり、解放を行うことにより使用される領域である。この確保や解放におけるメモリ領域の管理は、オペレーティングシステム（以下、OSと記す）が行なっている。従って、例えばアプリケーションは、OSが提供するシステムコールを用いてメモリの確保や解放を行っている。

【0008】

また、メモリは、コストなどを抑えるために、なるべく必要最小限の記憶容量で搭載されるのが一般的である。

【発明の開示】

【発明が解決しようとする課題】

【0009】

従って、上述したプログラムの更新のような、さほど頻度の多くない処理を行うためのメモリを予め融合機に搭載することは、コストに反映され好ましくない。その一方で、画像形成装置は、比較的大きなデータである画像データを扱うため、メモリの記憶容量は多ければ多いほど良い。

【0010】

本発明はこのような問題点に鑑み、画像形成装置に搭載されているメモリ資源を有効的に利用する画像形成装置、記憶領域確保方法を提供することを目的とする。

【課題を解決するための手段】

【0011】

上記課題を解決するために、本発明は、画像形成処理で使用するハードウェア資源と、画像形成に係る処理をプログラムに基づき実行するプロセスとを有する画像形成装置において、前記プロセスの処理の実行を制限させた状態であるオフライン状態に前記プロセスを移行させるオフライン手段と、前記オフライン状態に移行した前記プロセスが使用していた記憶領域を解放する記憶領域解放手段と、前記記憶領域解放手段が解放した記憶領域を、データを展開する記憶領域として使用するデータ展開手段とを有することを特徴とする。

【0012】

また、上記課題を解決するために、本発明は、前記データ展開手段は、前記記憶領域に前記データを展開する際に、前記オフライン手段により、前記プロセスをオフライン状態に移行させることを特徴とする。

【0013】

また、上記課題を解決するために、本発明は、前記データ展開手段は、前記オフライン手段が前記プロセスをオフライン状態に移行させると、前記記憶領域解放手段により、オフライン状態に移行した前記プロセスが使用していた記憶領域を解放させることを特徴とする。

【0014】

また、上記課題を解決するために、本発明は、前記オフライン手段は、前記プロセスに対し、前記プロセスをオフライン状態に移行させるオフライン移行要求を通知して、前記プロセスをオフライン状態に移行させることを特徴とする。

【0015】

また、上記課題を解決するために、本発明は、前記オフライン手段は、前記プロセスからの前記オフライン移行要求に対する応答に基づき、前記プロセスがオフライン状態に移行したかどうかを前記データ展開手段に通知することを特徴とする。

【0016】

また、上記課題を解決するために、本発明は、前記オフライン手段は、前記オフライン移行要求を通知した全てのプロセスがオフライン状態に移行すると、前記データ展開手段に対し、前記プロセスがオフライン状態に移行したことを通知することを特徴とする。

【0017】

また、上記課題を解決するために、本発明は、前記オフライン手段は、前記オフライン移行要求を通知した前記プロセスから、オフライン状態に移行できないことを通知されると、前記データ展開手段に対し、前記プロセスがオフライン状態に移行しなかったことを通知することを特徴とする。

【0018】

また、上記課題を解決するために、本発明は、前記オフライン手段は、前記オフライン移行要求を通知した前記プロセスのうち、1つ以上のプロセスから、オフライン状態に移行できないことを通知されても、前記オフライン移行要求を通知した全てのプロセスからの応答を待ったのちに、前記データ展開手段に対し、前記プロセスがオフライン状態に移行しなかったことを通知することを特徴とする。

【0019】

また、上記課題を解決するために、本発明は、前記オフライン手段は、前記オフライン移行要求を通知した前記プロセスのうち、1つのプロセスからオフライン状態に移行できないことを通知されると、前記オフライン移行要求を通知した他のプロセスからの応答を待たずに、前記データ展開手段に対し、前記プロセスがオフライン状態に移行しなかったことを通知することを特徴とする。

【0020】

また、上記課題を解決するために、本発明は、前記オフライン手段は、前記プロセスに前記オフライン移行要求を通知してから時間を計測し、前記オフライン移行要求に対する前記プロセスの応答が、所定の時間が経過しても通知されない場合、前記データ展開手段に対し、前記プロセスがオフライン状態に移行したことを通知することを特徴とする。

**【0021】**

また、上記課題を解決するために、本発明は、オフライン状態にされた前記プロセスは、前記オフライン手段がオフライン状態を解除することにより、制限されていた処理の実行が可能となることを特徴とする。

**【0022】**

また、上記課題を解決するために、本発明は、前記オフライン状態に移行したプロセスが実行を制限される処理は、他のプロセスからの要求により実行される処理であることを特徴とする。

**【0023】**

また、上記課題を解決するために、本発明は、前記オフライン状態に移行したプロセスは、他のプロセスからの要求を保持しておくことを特徴とする。

**【0024】**

また、上記課題を解決するために、本発明は、前記オフライン状態に移行したプロセスを消滅させるプロセス消滅手段をさらに有することを特徴とする。

**【0025】**

また、上記課題を解決するために、本発明は、前記プロセス消滅手段は、各プロセスに予め定められた順番で前記プロセスを消滅することを特徴とする。

**【0026】**

また、上記課題を解決するために、本発明は、前記順番は、各プロセスに予め定められた優先度に応じて定まることを特徴とする。

**【0027】**

また、上記課題を解決するために、本発明は、前記順番は、各プロセスに予め割り当てられた記憶領域のサイズに応じて定まることを特徴とする。

**【0028】**

また、上記課題を解決するために、本発明は、前記順番は、前記プロセスに割り当てられた記憶領域の位置に応じて定まることを特徴とする。

**【0029】**

また、上記課題を解決するために、本発明は、前記記憶領域解放手段は、前記プロセス消滅手段がプロセスを消滅させたことにより不要となった記憶領域を解放することを特徴とする。

**【0030】**

また、上記課題を解決するために、本発明は、前記記憶領域解放手段は、展開される前記データのサイズに応じて、前記記憶領域を解放することを特徴とする。

**【0031】**

また、上記課題を解決するために、本発明は、前記記憶領域解放手段は、前記記憶領域の解放を行うと、前記データ展開手段に対し、前記記憶領域の解放が完了したことを通知することを特徴とする。

**【0032】**

また、上記課題を解決するために、本発明は、前記データ展開手段が展開するデータは、前記プログラムを更新する更新プログラムであり、前記データ展開手段は、前記更新プログラムを通信手段を用いて取得することを特徴とする。

**【0033】**

また、上記課題を解決するために、本発明は、前記データ展開手段からのプログラム更新開始要求により、前記プログラムを更新するプログラム更新手段をさらに有することを特徴とする。

**【0034】**

また、上記課題を解決するために、本発明は、前記画像形成装置を操作するための入力手段を有し、前記プログラム更新手段は、前記プログラムを更新する際に、前記入力手段を無効化することを特徴とする。

**【0035】**



また、上記課題を解決するために、本発明は、前記プログラム更新手段は、前記プログラムの更新が終了すると、前記画像形成装置を再起動することを特徴とする。

【0036】

また、上記課題を解決するために、本発明は、前記プログラム更新手段は、前記画像形成装置と通信可能な機器に対し、前記プログラムの更新状況を通知することを特徴とする。

【0037】

また、上記課題を解決するために、本発明は、前記プログラム更新手段は、前記プログラムの更新状況を、オフライン状態に移行したプロセスを用いて通知することを特徴とする。

【0038】

また、上記課題を解決するために、本発明は、前記記憶領域は、前記プログラムの実行および前記ハードウェア資源を管理するオペレーティングシステムの管理外にあることを特徴とする。

【0039】

また、上記課題を解決するために、本発明は、画像形成処理で使用されるハードウェア資源と、画像形成に係る処理をプログラムに基づき実行するプロセスと、前記プログラムの実行および前記ハードウェア資源を管理するオペレーティングシステムとを有する画像形成装置の記憶領域確保方法であって、前記プロセスに対し、処理の実行を制限させるオフライン段階と、前記オペレーティングシステムの管理外にあり、前記オフライン段階で処理の実行が制限された前記プロセスが使用していた記憶領域を解放する記憶領域解放段階と前記記憶領域解放手段が解放した記憶領域に、データを展開するデータ展開段階とを有することを特徴とする。

【0040】

また、上記課題を解決するために、本発明は、前記記憶領域解放段階で、前記プロセスがオフライン状態に移行した後、前記プロセスを消滅させるプロセス消滅段階をさらに有することを特徴とする。

【0041】

また、上記課題を解決するために、本発明は、前記データ展開段階で展開されるデータは、前記プログラムを更新する更新プログラムであることを特徴とする。

【発明の効果】

【0042】

本発明によれば、画像形成装置に搭載されているメモリ資源を有効的に利用する画像形成装置、記憶領域確保方法を提供できる。

【発明を実施するための最良の形態】

【0043】

以下、本発明の実施の形態について図面に基づいて説明する。

【実施例】

【0044】

図1は、本発明による画像形成装置である融合機の一実施例の構成図を示す。融合機1は、ソフトウェア群2と、融合機起動部3と、ハードウェア資源4とを含むように構成される。

【0045】

融合機起動部3は融合機1の電源投入時に最初に実行され、アプリケーション層5およびプラットフォーム層6を起動する。例えば融合機起動部3は、アプリケーション層5およびプラットフォーム層6のプログラムをハードディスク装置（以下、HDDという）などから読み出し、読み出した各プログラムをメモリ領域に転送して起動する。ハードウェア資源4は、スキャナ11と、プロッタ12と、ファクシミリなどからなるその他のハードウェアリソース13を含む。

【0046】

また、ソフトウェア群 2 は、UNIX（登録商標）などのオペレーティングシステム（以下、OS という）上に起動されているアプリケーション層 5 とプラットホーム層 6 とを含む。アプリケーション層 5 は、プリンタ、コピー、ファックス、スキャナおよびネットファイルなどの画像形成にかかるユーザサービスにそれぞれ固有の処理を行うプログラムを含む。

【0047】

アプリケーション層 5 は、プリンタ用のアプリケーションであるプリンタアプリ 21 と、コピー用アプリケーションであるコピーアプリ 22 と、ファックス用アプリケーションであるファックスアプリ 23 と、スキャナ用アプリケーションであるスキャナアプリ 24 と、ネットワークを用いて文書のサービスを行うネットファイルアプリ 25 を含む。

【0048】

また、プラットホーム層 6 は、アプリケーション層 5 からの処理要求を解釈してハードウェア資源 4 の獲得要求を発生するコントロールサービス層 9 と、1 つ以上のハードウェア資源 4 の管理を行ってコントロールサービス層 9 からの獲得要求を調停するシステムリソースマネージャ（以下、SRM という）39 と、SRM 39 からの獲得要求に応じてハードウェア資源 4 の管理を行うハンドラ層 10 とを含む。

【0049】

コントロールサービス層 9 は、ネットワークコントロールサービス（以下、NCS という）31、デリバリーコントロールサービス（以下、DCS という）32、オペレーションパネルコントロールサービス（以下、OCS という）33、ファックスコントロールサービス（以下、FCS という）34、エンジンコントロールサービス（以下、ECS という）35、メモリコントロールサービス（以下、MCS という）36、ユーザインフォメーションコントロールサービス（以下、UCS という）37、システムコントロールサービス（以下、SCS という）38、リモートサービス（以下、RS という）42、オンデマンドアップデートサービス（以下、OUS という）43 など、一つ以上のサービスモジュールを含むように構成されている。

【0050】

なお、プラットホーム層 6 は予め定義されている関数により、アプリケーション層 5 からの処理要求を受信可能とする API 53 を有するように構成されている。OS は、アプリケーション層 5 およびプラットホーム層 6 の各ソフトウェアをプロセスとして並列実行する。

【0051】

NCS 31 のプロセスは、ネットワーク I/O を必要とするアプリケーションに対して共通に利用できるサービスを提供するものであり、ネットワーク側から各プロトコルによって受信したデータを各アプリケーションに振り分けたり、各アプリケーションからのデータをネットワーク側に送信する際の仲介を行う。

【0052】

例えば NCS 31 は、ネットワークを介して接続されるネットワーク機器とのデータ通信を h t t p d（HyperText Transfer Protocol Daemon）により、H T T P（HyperText Transfer Protocol）で制御する。

【0053】

また、NCS 31 は、更新プログラムをネットワークからダウンロードするためにも用いられる。

【0054】

DCS 32 のプロセスは、蓄積文書の配信などの制御を行う。OCS 33 のプロセスは、オペレータと本体制御との間の情報伝達手段となるオペレーションパネルの制御を行う。また、このオペレーションパネルは、オペレータが融合機 1 を操作するための入力手段でもある。FCS 34 のプロセスは、アプリケーション層 5 から P S T N または I S D N 網を利用したファックス送受信、バックアップ用のメモリで管理されている各種ファックスデータの登録／引用、ファックス読み取り、ファックス受信印刷などを行うための A P

I を提供する。

【0055】

ECS35のプロセスは、スキャナ11、プロッタ12、その他のハードウェアリソース13などのエンジン部の制御を行う。MCS36のプロセスは、メモリの取得および解放、HDDの利用などのメモリ制御を行う。UCS37のプロセスは、ユーザ情報の管理を行うものである。RS42のプロセスは、リモートサービスの制御を行うものである。OUS43のプロセスは、ダウンロードした更新プログラムをメモリに展開するなど、ダウンロードに関する種々の処理を行うものである。

【0056】

SCS38のプロセスは、アプリケーション管理、操作部制御、システム画面表示、LED表示、ハードウェア資源管理、割り込みアプリケーション制御などの処理を行う。

【0057】

SRM39のプロセスは、SCS38と共にシステムの制御およびハードウェア資源4の管理を行うものである。例えばSRM39のプロセスは、スキャナ11やプロッタ12などのハードウェア資源4を利用する上位層からの獲得要求に従って調停を行い、実行制御する。

【0058】

具体的に、SRM39のプロセスは獲得要求されたハードウェア資源4が利用可能であるか（他の獲得要求により利用されていないかどうか）を判定し、利用可能であれば獲得要求されたハードウェア資源4が利用可能である旨を上位層に通知する。また、SRM39のプロセスは上位層からの獲得要求に対してハードウェア資源4を利用するためのスケジューリングを行い、要求内容（例えば、プリンタエンジンによる紙搬送と作像動作、メモリ確保、ファイル生成など）を直接実施している。

【0059】

また、ハンドラ層10は後述するファックスコントロールユニット（以下、FCUという）の管理を行うファックスコントロールユニットハンドラ（以下、FCUHという）40と、プロセスに対するメモリの割り振り及びプロセスに割り振ったメモリ領域の管理を行う記憶領域管理手段であるイメージメモリハンドラ（以下、IMHという）41とを含む。SRM39およびFCUH40は、予め定義されている関数によりハードウェア資源4に対する処理要求を送信可能とするエンジンI/F54を利用して、ハードウェア資源4に対する処理要求を行う。

【0060】

融合機1は、各アプリケーションで共通的に必要な処理をプラットフォーム層6で一元的に処理することができる。次に、融合機1のハードウェア構成について説明する。

【0061】

図2は、融合機1の一実施例のハードウェア構成図を示している。融合機1は、コントローラボード60と、オペレーションパネル52と、FCU68と、エンジン79とを含む。また、FCU68は、G3規格対応ユニット69と、G4規格対応ユニット70とを有する。

【0062】

また、コントローラボード60は、CPU61と、ASIC66と、HDD65と、システムメモリ（MEM-P）63と、ローカルメモリ（MEM-C）64と、フラッシュメモリ80と、ノースブリッジ（以下、NBと記す）62と、サウスブリッジ（以下、SBと記す）73と、NIC174（Network Interface Card）と、USBデバイス175と、IEEE1394デバイス176と、セントロニクスデバイス177とを含む。

【0063】

オペレーションパネル52は、コントローラボード60のASIC66に接続されている。また、SB173と、NIC174と、USBデバイス175と、IEEE1394デバイス176と、セントロニクスデバイス177は、NB62にPCIバスで接続されている。

**【0064】**

また、FCU68と、エンジン79と、コントローラボード60のASIC66にPCIバスで接続されている。

**【0065】**

なお、コントローラボード60は、ASIC66にローカルメモリ64、HDD65などが接続されると共に、CPU61とASIC66とがCPUチップセットのNB62を介して接続されている。このように、NB62を介してCPU61とASIC66とを接続すれば、CPU61のインタフェースが公開されていない場合に対応できる。

**【0066】**

また、ASIC66とNB62とはPCIバスを介して接続されているのではなく、AGP (Accelerated Graphics Port) 67を介して接続されている。このように、図2のアプリケーション層5やプラットフォーム層6を形成する一つ以上のプロセスを実行制御するため、ASIC66とNB62とを低速のPCIバスでなくAGP67を介して接続し、パフォーマンスの低下を防いでいる。

**【0067】**

CPU61は、融合機1の全体制御を行うものである。CPU61は、NCS31、DCS32、OCS33、FCS34、ECS35、MCS36、UCS37、RS42、OUS43、SCS38、SRM39、FCUH40、IMH41、MEU44をOS上にそれぞれプロセスとして起動して実行させると共に、アプリケーション層5を形成するプリンタアプリ21、コピーアプリ22、ファックスアプリ23、スキャナアプリ24、ネットファイルアプリ25を起動して実行させる。

**【0068】**

NB62は、CPU61、システムメモリ63、SB173およびASIC66を接続するためのブリッジである。システムメモリ63は、融合機1の描画用メモリなどとして用いるメモリである。SB173は、NB62とPCIバス、周辺デバイスとを接続するためのブリッジである。また、ローカルメモリ64はコピー用画像バッファ、符号バッファとして用いるメモリである。フラッシュメモリ80は、プログラムが格納されるメモリである。

**【0069】**

ASIC66は、画像処理用のハードウェア要素を有する画像処理用途向けのICである。HDD65は、画像の蓄積、文書データの蓄積、プログラムの蓄積、フォントデータの蓄積、フォームの蓄積などを行うためのストレージである。また、オペレーションパネル52は、ユーザからの入力操作を受け付けると共に、ユーザに向けた表示を行う操作部である。

**【0070】**

次に、記憶領域を確保する処理を説明するために、更新プログラムをダウンロードする処理を例にして説明するが、その説明に先立ち、以下の説明で用いられるプロセスやアプリケーションなどの説明をしておく。

**【0071】**

本実施の形態においては、アプリケーション、モジュール、プロセスという表現が用いられている。このうちのプロセスは、一般的に実行中のプログラムを表すため、アプリケーションやモジュールはプロセスに含まれる。したがって、いずれの表現の場合も本質的な差はないが、本実施の形態では、説明内容に応じて適宜使い分ける。

**【0072】**

また、本実施の形態において、ダウンロードして取得したプログラムを更新プログラムとしているが、この更新プログラムは、融合機1が有する従来からのプログラム全体を更新するプログラムだけではなく、従来のプログラムとマージするプログラムや従来のプログラムの一部を更新する差分プログラムを含む。

**【0073】**

以下、メモリ領域を確保し、更新プログラムをダウンロードする処理を説明する。

**【0074】**

融合機1は、例えば融合機1の開発元などが提供するサーバから、プログラムの更新を開始する通知を受けると、プロセスが使用していたメモリ領域を解放することにより、更新プログラムを展開するためのメモリ領域を確保し、そのメモリ領域に更新プログラムを展開するという処理を行う。そこでまず、メモリ領域についての説明をする。

**【0075】**

メモリ領域の内容は、通常、メモリマップを用いて示される。このメモリマップとは、どの位置に何の情報を記憶するのかが示されたものである。

**【0076】**

このメモリマップを用いてメモリ領域について説明する。図3には、揮発性の記憶装置であるシステムメモリ62のメモリマップ71と、メモリマップ71に示されている共有メモリを拡大したメモリマップ72とメモリ領域VM75とが示されている。これらのメモリマップ71、72に表示されているメモリ領域は、メモリ領域VM75を除き、IMH41が管理している。メモリ領域VM75は、OSが管理するメモリ領域である。

**【0077】**

このようにOSの管理外にあるメモリ領域を用いることにより、融合機1を開発する側では、融合機1の仕様や特性に合わせたメモリ管理を行えるばかりではなく、メモリリークなど、不具合の発生原因を抑制することが可能となる。

**【0078】**

なお、以下の説明では、上述したOSの管理外にあるメモリ領域を用いた処理の説明となるが、OSが管理するメモリ領域でも同様の処理は可能である。

**【0079】**

メモリマップ71の説明をする。メモリマップ71に示されるように、システムメモリ62は、共有メモリ、プリンタエミュレーション作業用メモリ、ページメモリ、印字データ用、その他、というようにメモリ領域が分けられている。

**【0080】**

このうち、共有メモリは、実行しているプログラムが展開されるとともに、プロセスが処理を行うために使用するメモリである。図3に示されているメモリマップ72は、プログラムが展開する前の状態を示しているため、全て空き領域となっている。

**【0081】**

この状態から、プロセスが起動することにより、メモリマップは、図4に示されるようになる。図4に示されるメモリマップは、図3で空きだった共有メモリに、アプリケーションやモジュールが展開された様子を示すものである。このメモリマップ73には、コピーアプリ、プリンタアプリなどのアプリケーションやサービス層のモジュールで利用される領域が、どの位置に展開されるのかが示されている。

**【0082】**

次に示す図5は、図4に示されるようにメモリ領域に展開されたプロセスの稼働状況を示す図である。図5に示される各プロセスの稼働状況は、プロセスを示す四角内に表示され、この図の場合、全てのプロセスが「通常」の処理を行なっている状況である。

**【0083】**

この稼働状況で、例えば、図6に示されるように、各アプリケーションや、サービス層モジュールの一部などのプロセスをオフライン状態にさせたとする。このオフライン状態とは、プロセスが、処理の実行を制限された状態であることを示している。具体的には、エンジン状態通知や、異常発生のお知らせなどを受信することは可能であるが、対外的なイベントに対する実行動作を禁止した状態である。

**【0084】**

具体的に説明すると、それぞれのアプリケーションや、サービス層モジュールは、オフライン状態になると、図7に示される処理の実行を制限する。例えば、コピーアプリ22は、読み取り・印刷・LS蓄積に係る処理を行わない。FAXアプリ23は、送受信・LS蓄積・受信文書印刷に係る処理を行わない。プリンタアプリ21は、印刷・LS蓄積に

係る処理を行わない。スキャナアプリ 24 は、読み取り・LS 蓄積に係る処理を行わない。ネットファイルアプリ 25 は、印刷・LS 蓄積に係る処理を行わない。SCS 38 は、操作部イベントをマスクし、ジョブの実行要求を拒絶する。NC S 31 は、通信イベントを拒絶する。

#### 【0085】

このようなオフライン状態を解除することは可能であり、その場合、プロセスは、制限されていた処理を実行することが可能となる。したがって、アプリケーションは、オフライン状態で動作している際にも、他のアプリケーションなどから通知された内容は保持しておき、処理の実行の制限を解除されたときに、保持していた内容に基づき処理を再開することが可能となっている。

#### 【0086】

このように、オフライン状態における処理の実行の制限とは、その制限を解除することにより、制限されていた処理が再び実行可能となる範囲での制限である。したがって、制限する処理は、その範囲内で適宜定めることができる。

#### 【0087】

プロセスをオフライン状態にさせたのち、SCS 38 は、プロセスが使用していたメモリ領域を解放させる。ここで、プロセスが使用していたメモリ領域について、図 8 のメモリマップ 74 を用いて説明する。メモリマップ 74 は、プロセスが使用していたメモリ領域を解放したメモリ領域の様子を示すメモリマップである。

#### 【0088】

メモリマップ 73 (図 4 参照) に示したように、それぞれのアプリケーションや、モジュールには、システムメモリ 63 上に領域が割り当てられている。その割り当てられたメモリ領域は、さらに 2 つの領域で構成される。これらのメモリ領域を本実施の形態では、1 つを固定エリアとし、もう 1 つを変動エリアとする。このうち、固定エリアとは、エンジン 79 (図 2 参照) の状態通知を記憶しておくなど、各アプリケーションがジョブを実行するしないにかかわらず必要とする情報を記憶しておくエリアである。これらの情報のうち、例えばエンジン 79 の状態が記憶されていない場合、オフライン状態が解除されて再びプロセスが処理を実行した際に、実際のエンジン 79 の状態が把握できていないため、異常動作の原因となる可能性がある。

#### 【0089】

また、変動エリアとは、プロセスが使用していたメモリ領域であり、各アプリケーションやモジュールが処理を実行する際に必要となるメモリ領域である。このメモリ領域が使用される例として画像情報の一時的な記憶や、現在の動作に関する情報の記憶に用いられる。

#### 【0090】

このように、本実施の形態においては、上記のように変動エリアを定めているが、これはあくまでも一例であり、上述した処理の制限の範囲によって種々の定め方が存在する。

#### 【0091】

図 8 の説明に戻り、上述した変動エリアが解放されると、図 8 に示されるような空きのメモリ領域が得られる。

#### 【0092】

次に、固定エリアをさらに解放させるためのプロセスの消滅について説明する。一般的に、ある処理を実行する場合、OS は、そのプログラムをメモリに展開し、プロセスとして登録して処理を実行させ、必要がなくなると、OS は、プロセスとしての登録を抹消する。

#### 【0093】

プロセスの消滅とは、本実施の形態において、プロセスとしての登録抹消を意味する。従って、プロセスが消滅させられると、展開されたプログラムを含むプロセスに割り当てられた領域は、実質的に空きとなる。本実施の形態の場合、このプロセスに割り当てられた領域が、上述した固定エリアである。

**【0094】**

プロセスを消滅させると、図5に示される稼働状況から図9に示される稼働状況となる。図9に示される稼働状況は、全てのアプリケーションを消滅させた場合の稼働状況を示す図であり、消滅したアプリケーションが点線で示されている。このとき、図10のメモリマップ78に示されるように、解放しなかったサービス層モジュールに割り当てられているメモリ領域以外は、全て空き領域となる。

**【0095】**

このようにして得られた空き領域を、ダウンロードした更新プログラムを展開するメモリ領域として用いることで、更新プログラムをダウンロードするためのメモリ領域を予め用意しなくても更新プログラムをダウンロードすることが可能となる。

**【0096】**

また、ダウンロードした更新プログラムは、新たなプログラムとしてフラッシュメモリ69に格納される。

**【0097】**

次に、プロセスの消滅に伴う処理について、SCS38とアプリケーションを例にして説明する。図11には、SCS38のプロセス201とアプリケーションを構成するプロセス204とが示されている。そして、SCS38のプロセス201には、SCS38のメインタスク202とアプリエラーハンドラ203とが示されている。また、アプリケーションを構成するプロセス204には、アプリメインタスク205とSCSエラーハンドラ206とが示されている。

**【0098】**

プロセス201と、プロセス204は、図11に示されるように、起動時に互いにエラーハンドラを登録する。これらエラーハンドラは、この場合、互いの存在を確認するものである。したがって、図12に示されるように、プロセス204が消滅すると、SCS38のアプリエラーハンドラ203が、プロセス204が消滅したことを認識し、エラーが発生することになる。

**【0099】**

そこで、SCS38は、エラーを回避するため、図13に示されるように、予めアプリエラーハンドラを停止してからアプリケーションのプロセスを消滅する。このようにすることで、SCS38は、アプリケーションのプロセスが消滅しても、アプリエラーハンドラは停止しているため、エラーの発生を回避することができる。なお、図中に示される「KILL」は、プロセスを消滅させるコマンドを意味する。

**【0100】**

このように、プロセスを消滅させることで、そのプロセスに割り当てられた固定エリアと変動エリアは、実質的に空きとなる。従って、SCS38は、固定エリアと変動エリアを解放させることが可能となる。

**【0101】**

このようにすることにより、応用ソフトウェアが管理を行うメモリ領域に、より多くのメモリ領域を確保することが可能となる。

**【0102】**

そして、得られた空き領域は、ダウンロードした更新プログラムを展開するメモリ領域として用いられる。また、ダウンロードした更新プログラムは、新たなプログラムとしてフラッシュメモリ80に格納される。

**【0103】**

以下、メモリ領域を解放し、ダウンロードする処理について説明する。上述したように、メモリ領域の解放には、プロセスをオフライン状態して変動エリアを解放する場合と、プロセスを消滅することで固定エリアと変動エリアを解放する場合の2通りの処理がある。そのため、それぞれ分けて説明することとする。

**【0104】**

まず、プロセスをオフライン状態して変動エリアを解放する場合の処理について説明す

る。

#### 【0105】

図14は、融合機1が2つのアプリケーション（アプリA81、アプリB82）をオフライン状態にしてメモリ領域を解放し、更新プログラムをダウンロードし、フラッシュメモリ69に格納されたプログラムを更新する処理を示すシーケンス図である。この処理は、通信手段に対応するNCS31と、データ展開手段に対応するOUS43と、記憶領域解放手段とオフライン手段とプロセス消滅手段とプログラム更新手段に対応するSCS38と、エンジンファーム55により行われる。なお、エンジンファームとは、エンジンに搭載されたファームウェアを示す。

#### 【0106】

以下、シーケンス図の説明をする。ステップS101で、NCS31は、プログラム更新開始をネットワークから通知され、ステップS102で、OUS43に、プログラムの更新開始を示すプログラム更新開始を通知する。通知を受けたOUS43は、ステップS103で、SCS38に対し、アプリオフライン要求を通知する。このアプリオフライン要求とは、アプリケーションを上記したオフライン状態にさせる要求である。

#### 【0107】

また、このアプリオフライン要求は、SCS38を介してアプリケーションに通知されるが、この要求を通知されたアプリケーションは、アプリケーションとしての動作を行っていない限り、オフライン状態に移行しなければならない。

#### 【0108】

この「アプリケーションとしての動作」とは、例えばコピーアプリ22を例にして説明すると、コピーの処理を実行する動作をいう。

#### 【0109】

従って、コピーアプリ22がコピーの処理を実行していない限り、コピーアプリ22は、オフライン状態で動作しなければならない。

#### 【0110】

シーケンス図の説明に戻り、アプリオフライン要求をOUS43から通知されたSCS38は、アプリA81と、アプリB82にステップS104とステップS105でアプリオフライン移行要求を通知する。

#### 【0111】

アプリオフライン移行要求を通知されたアプリA81、アプリB82は、それぞれオフライン状態になることが可能である場合、SCS38に対し、オフライン状態に移行したことを示すアプリオフライン応答（OK）を通知する。このシーケンス図の場合、アプリA81、アプリB82ともにオフライン状態になることが可能であるため、アプリA81、アプリB82は、ステップS106、ステップS107で、それぞれアプリオフライン応答（OK）をSCS38に通知している。

#### 【0112】

アプリオフライン応答（OK）を通知されたSCS38は、ステップS108で、OUS43に対し、アプリケーションがオフライン状態になったことを示すアプリオフライン確定を通知する。

#### 【0113】

アプリケーションのオフライン確定を通知されたOUS43は、ステップS109でSCS38にオフライン状態に移行したアプリケーションの変動エリアを解放させるためにメモリ解放要求を通知する。

#### 【0114】

メモリ解放要求を通知されたSCS38は、アプリA81と、アプリB82に対し、ステップS110とステップS111でメモリ解放要求を通知する。メモリ解放要求を通知されたアプリA81、アプリB82は、それぞれメモリを解放し、ステップS112、ステップS113で、SCS38にメモリ解放完了応答を通知する。

#### 【0115】



メモリ解放完了応答を通知されたSCS38は、OUS43にステップS114でメモリ領域の解放が完了したことを示すメモリ解放完了応答を通知する。メモリ解放完了応答を受信したOUS43は、更新プログラムをダウンロードし、メモリ領域の解放により得られたメモリ領域に更新プログラムを展開する。

#### 【0116】

そして、OUS43は、ステップS115で、プログラム更新開始要求であるリモートROM更新開始要求をSCS38に通知する。リモートROM更新開始要求を通知されたSCS38は、キーイベントのマスクを行う。このキーイベントのマスクとは、オペレーションパネル52に表示されるキーが押下されたことにより発生するイベントを受け付けず、オペレーションパネル52を無効化する処理である。

#### 【0117】

次に、SCS38は、プログラムの更新処理を行うために、ステップS116で、ダウンロードした更新プログラムをエンジンファーム55に送信する。エンジンファーム55は、プログラムの更新を行い、その結果をステップS117でSCS38に通知する。

#### 【0118】

SCS38は、プログラムを更新した結果が通知されると、ステップS118で、エンジンファーム55に対し、リポート準備要求を行う。そして、ステップS119で、エンジンファーム55からリポート準備完了を通知されると、SCS38は、強制リポートを行い、プログラム更新処理を終了する。

#### 【0119】

以上のように、SCS38は、アプリケーションが使用するメモリ領域を解放し、OUS43は、プログラムを更新する更新プログラムを、NCS31を用いてダウンロードし、SCS38により解放されたメモリ領域に展開する。

#### 【0120】

以上説明したフローチャートにおけるステップS103からステップS108が、オフライン段階に対応し、ステップS109からステップS114が、記憶領域解放段階に対応し、ステップS114の後に実行されるOUS43のダウンロードプログラムのメモリ展開が、データ展開段階に対応する。

#### 【0121】

次に、OUS43と、SCS38と、アプリケーションの処理の詳細を、フローチャートを用いて説明する。

#### 【0122】

最初に、OUS43の処理について、図15のフローチャートを用いて説明する。ステップS201で、OUS43は、NCS31からプログラムの更新開始を示すプログラム更新開始通知を受信する。次のステップS202で、OUS43は、受信したプログラム更新開始通知自体が正常なデータであるかどうかをチェックするためのSUM値のチェックを実行し、プログラム更新開始通知が正常データでない場合は、ステップS203で、プログラムの更新処理を終了する。

#### 【0123】

ステップS202で、プログラム更新開始通知が正常なデータであると判断された場合、OUS43は、ステップS204で、SCS38にアプリケーションをオフライン状態にさせるアプリオフライン要求を通知する。アプリオフライン要求の応答であるアプリオフライン確定をステップS205で受信すると、OUS43は、ステップS206で、アプリオフラインが実行完了したかどうか判断する。このステップS206の処理は、アプリオフラインが確定しなかったことを通知された場合に、ステップS203で、プログラムの更新処理を終了するための処理である。

#### 【0124】

ステップS206で、OUS43が、アプリオフラインの実行が完了したと判断した場合、ステップS207で、SCS38にメモリ解放要求を行う。そして、ステップS208で、OUS43は、SCS38からメモリ解放完了を受信すると、ステップS209で

更新プログラムの展開に必要なメモリ領域が確保できたかどうか判断する。

【0125】

OUS43が、メモリ領域が確保できなかったと判断した場合、OUS43は、ステップS210で、リブートを実行するかどうか判断する。OUS43が、リブートを実行しないと判断した場合、OUS43は、オフライン状態を解除するためにステップS211で、SCS38にオフライン解除要求を通知し、ステップS212でプログラム更新処理を終了する。

【0126】

ステップS210で、OUS43はリブートを実行すると判断した場合、OUS43は、融合機1を再起動させるためにSCS38にリブート要求を通知し、処理を終了する。

【0127】

ステップS209の処理に戻り、プログラムの展開に必要なメモリを確保できた場合、OUS43は、ステップS214でダウンロードプログラムをメモリに展開する。そして、ステップS215で、OUS43は、プログラム更新開始要求であるリモートROM更新開始要求をSCS38に通知し、処理を終了する。

【0128】

次に、SCS38の処理について、図16のフローチャートを用いて説明する。ステップS301で、SCS38は、OUS43からアプリオフライン要求を受信する。アプリオフライン要求を受けたSCS38は、ステップS302で、各アプリケーションにアプリオフライン移行要求を通知する。

【0129】

次のステップS303で、SCS38は、全てのアプリケーションからの応答が、アプリオフライン応答（OK）ではなかった場合、ステップS304で、各アプリケーションに、アプリオフライン解除を通知する。これにより、アプリケーションは、オフライン状態が解除され、制限されていた処理の実行が可能となる。次に、SCS38は、ステップS305でOUS43に対し、アプリケーションがオフラインに移行しなかったことを、アプリオフライン確定として通知し、処理を終了する。

【0130】

ステップS303で、全てのアプリケーションからの応答が、アプリオフライン応答（OK）である場合、SCS38は、ステップS306でOUS43にアプリオフライン確定を通知する。

【0131】

次に、SCS38は、ステップS307で、OUS43からメモリ解放要求を受信すると、ステップS308でメモリ領域の解放処理を実行し、ステップS309でOUS43にメモリ解放完了を通知する。

【0132】

OUS43にメモリ解放完了を通知すると、SCS38は、OUS43の要求により処理が分岐する。まず、OUS43からオフライン解除要求を受信した場合、SCS38は、ステップS310からステップS311に分岐し、各アプリケーションアプリオフライン解除を通知し、処理を終了する。

【0133】

ステップS310で、SCS38がOUS43からオフライン解除要求を受信しなかった場合、SCS38は、ステップS312で、OUS43からリモートROM更新開始要求を受信したかどうか判断する。SCS38がリモートROM更新開始要求を受信した場合、SCS38は、ステップS315で、オペレーションパネル52上のキー・タッチパネル操作イベントをマスクしてキーイベントのマスクを行う。

【0134】

このように、プログラムを更新する際に、オペレーションパネル52を無効化することにより、更新処理を迅速に行うことができるとともに、制限されていない機能は実行することができるため、例えば重要な機能をプログラム更新中にもかかわらず実行することが

できる。

#### 【0135】

次に、SCS38は、ステップS316で、プログラム更新処理を実行する。そして、SCS38は、ダウンロードの実行を開始したことやプログラムの更新状況などを、RS42からネットワークを通してサービスセンタなどに通知し、ステップS318でリブートを実行し処理を終了する。なお、サービスセンタとは、例えば製造元に備えられた融合機1の保守や点検を行うためのサーバや、更新プログラムの配布元に備えられたサーバなどである。また、各アプリケーションやモジュールの状態情報をサービスセンタに送信してもよいし、サービスセンタではそのログを記録しておいてもよい。

#### 【0136】

ステップS312で、リモートROM更新開始要求を受信しなかった場合、ステップS313で、SCS38は、OUS43からリブート要求を受信したかどうか判断する。SCS38がリブート要求を受信しなかった場合、SCS38は、再びステップS310の処理を行う。SCS38がリブート要求を受信した場合、SCS38は、ステップS314でダウンロードの実行が失敗し、リブート処理を行うことをRS42からネットワークを通してサービスセンタなどに通知し、ステップS318でリブートを行い処理を終了する。

#### 【0137】

次に、アプリA81、アプリB82などのアプリケーションの処理について、図17のフローチャートを用いて説明する。ステップS401は、アプリケーションの起動を示す。次のステップS402で、アプリケーションは、上述した固定エリア用のメモリ領域を確保する。次に、ステップS403で、アプリケーションは、変動エリア用のメモリ領域を確保し、通常の処理を行う状態となる。

#### 【0138】

そして、ステップS404で、SCS38は、SCS38からオフライン要求を受信すると、ステップS405で、オフライン状態に移行することが可能かどうか判断する。オフライン状態に移行することが不可能であれば、アプリケーションは、再びステップS404の処理に戻る。オフライン状態に移行することが可能であれば、アプリケーションは、ステップS406で、オフライン状態に移行する。

#### 【0139】

オフライン状態に移行したアプリケーションは、ステップS407で、SCS38からメモリ解放要求を受信すると、ステップS408で変動エリア用のメモリ領域を解放する。メモリ領域を解放したアプリケーションは、ステップS409で、SCS38からオフライン解除要求を受信すると、ステップS410で、変動エリア用のメモリ領域を確保し、制限されていた処理を実行することが可能となる。

#### 【0140】

次に、アプリケーションが、オフライン状態に移行可能かどうかを判断する処理を図18のフローチャートを用いて説明する。ステップS501で、アプリケーションは、SCS38からアプリオフライン移行要求を受信する。アプリケーションは、アプリオフライン移行要求を受信したことにより、所定の動作を実行中かどうかの判断を次のステップS502からステップS506まで行う。

#### 【0141】

このステップS502からステップS506までの動作のチェックは、印刷動作実行中、スキャン動作実行中、FAX送受信動作実行中、ネットワーク通信動作実行中の順に行われ、1つでも実行中であれば、ステップS503で、アプリケーションは、SCS38にオフライン状態に移行できないことを示すアプリオフライン応答(NG)を通知し、処理を終了する。

#### 【0142】

ステップS502からステップS506までのチェックにより、いずれも実行中でないと判断された場合、ステップS507でアプリケーションは、オフライン状態に移行する

。そして、ステップS508で、アプリケーションは、SCS38にアプリオフライン応答（OK）を通知する。

【0143】

そして、オフライン状態に移行したアプリケーションは、SCS38からの要求待ちとなる。その要求待ちの状態において、アプリケーションは、ステップS509でSCS38からメモリ解放要求を受信すると、ステップS510でメモリ解放処理を実行し、再び要求待ち状態となる。

【0144】

アプリケーションは、ステップS511で、SCS38からオフライン解除要求を受信すると、ステップS512でオフライン状態を解除する。

【0145】

次に、オフライン状態に移行した場合のアプリケーションのイベントに対する処理を図19のフローチャートを用いて説明する。ステップS601からステップS604はイベントの発生であり、それぞれ、印刷イベント、スキャナイベント、FAXイベント、ネットワーク通信イベントが発生している。

【0146】

次に、アプリケーションは、ステップS605で、オフライン状態かどうか判断する。アプリケーションがオフライン状態であれば、アプリケーションは、ステップS610からステップS613で、発生したイベントに対する処理の実行を拒絶する。

【0147】

ステップS605で、アプリケーションがオフライン状態であれば、ステップS606からステップS609で、発生したイベントに対する処理を実行する。

【0148】

次に、上述した処理におけるアプリケーション内部の処理を、スキャナアプリ24を例にして説明する。図20は、スキャナアプリ24の内部構造の概略図である。この概略図には、SCS38とスキャナアプリ24とが示されており、さらにスキャナアプリ24内のプロセスであるSYSOPプロセス85と、配信プロセス83と、TWA INプロセス84も示されている。

【0149】

SYSOPプロセス85は、外部のプロセスとのやり取りを行うプロセスであり、外部からの通知を適当なプロセスに振り分ける。配信プロセス83は、画像データなどをネットワークを介して配信するプロセスである。TWA INプロセス84は、画像入力装置用APIであるTWA INに関連する処理を行うプロセスである。

【0150】

次に、これらのプロセス間で行われる処理を図21のシーケンス図を用いて説明する。図21のシーケンス図は、SCS38からアプリオフライン要求を受信したスキャナアプリ24が、オフライン応答を返すまでのスキャナアプリ24内の処理を示している。

【0151】

ステップS701で、SCS38は、スキャナアプリ24のSYSOPプロセス85にアプリオフライン要求を通知する。アプリオフライン要求を通知されたSYSOPプロセス85は、配信プロセス83とTWA INプロセス84に、ステップS702とステップS703で状態要求を通知する。状態要求を通知された配信プロセス83とTWA INプロセス84は、それぞれステップS704とステップS705で、状態応答をOKまたはNGとしてSYSOPプロセス85に通知する。

【0152】

状態応答を通知されたSYSOPプロセス85は、配信プロセス83とTWA INプロセス84からの状態応答が、ともにOKである場合、ステップS706でSCS38にオフライン応答（OK）を通知する。配信プロセス83とTWA INプロセス84からの状態応答で1つ以上NGを通知された場合、SYSOPプロセス85は、ステップS706でSCS38にオフライン応答（NG）を通知する。

**【0153】**

このように、他のアプリケーションも、アプリケーションを構成するプロセスが1つでもNGであれば、そのアプリケーションは、SCS38にオフライン応答（NG）を行うようになっている。

**【0154】**

次に、オフライン状態の解除について、図22を用いて説明する。ステップS801で、SCS38は、スキャナアプリ24のSYSOPプロセス85にアプリオフライン解除要求を通知する。アプリオフライン解除要求を通知されたSYSOPプロセス85は、配信プロセス83とTWA INプロセス84に、ステップS802とステップS803で解除要求を通知する。解除要求を通知された配信プロセス83とTWA INプロセス84は、それぞれステップS804とステップS805で、常にOKである状態応答をSYSOPプロセス85に通知する。

**【0155】**

OKである状態応答を通知されたSYSOPプロセス85は、ステップS806でSCS38にオフライン解除応答（OK）を通知する。

**【0156】**

次に、スキャナアプリ24が、例えば画像データを配信する配信要求を通知された場合に実行される処理を、図23を用いて説明する。この図23は、スキャナアプリ24の処理とその関係を示した図であり、以下番号に従って説明する。ステップS1でスキャナアプリ24は、スキャナアプリ24の外部のプロセスから、配信要求を通知される。ステップS2とステップS3でスキャナアプリ24は、要求解析処理において、スキャナアプリ24の状態の判断処理を行う。

**【0157】**

その結果、状態がオフライン状態の場合、ステップS4とステップS5でスキャナアプリ24は、要求解析処理において、配信データを一時記憶する処理を行う。

**【0158】**

また、オフライン状態ではない場合、スキャナアプリ24は、通常の状態であるので、ステップS6とステップS7で配信処理を実行する。

**【0159】**

そして、ステップS8とステップS9で、スキャナアプリ24は、終了処理あるいは、オフライン状態では実行できない要求についてのエラー処理を行い、処理を終了する。

**【0160】**

このように、オフライン状態のアプリケーションは、外部からの要求を保持または実行不可能な要求であればエラー処理を行う。

**【0161】**

次に、アプリケーションがオフライン状態に移行できない場合の処理について説明する。

**【0162】**

図24は、SCS38が、1つ以上のアプリケーションからオフライン状態に移行できないことを通知されても、即座にOUS43にその旨を通知せず、全てのアプリケーションからの応答を待ったのちにオフライン状態に移行しなかったことを通知する処理を示すシーケンス図である。

**【0163】**

以下、シーケンス図の説明をする。ステップS901で、NCS31は、プログラム更新開始をネットワークから通知され、ステップS902で、OUS43に、プログラムの更新開始を示すプログラム更新開始を通知する。通知を受けたOUS43は、ステップS903で、SCS38に対し、アプリオフライン要求を通知する。

**【0164】**

オフライン要求をOUS43から通知されたSCS38は、アプリA81と、アプリB82にステップS904とステップS905でアプリオフライン移行要求を通知する。

**【0165】**

アプリオフライン移行要求を通知されたアプリB82は、オフライン状態に移行することができないため、ステップS906で、SCS38にアプリオフライン応答（NG）を通知する。また、アプリA81はオフライン状態に移行することが可能なため、ステップS907でアプリオフライン応答（OK）を通知する。

**【0166】**

次に、SCS38は、アプリA81と、アプリB82に、ステップS908とステップS909でアプリオフライン解除要求を通知する。そして、SCS38は、OUS43に対し、ステップS910でアプリケーションがオフラインに移行できなかったことを示すアプリオフライン確定（オフライン移行NG）を通知する。

**【0167】**

このように、SCS38は、オフライン移行要求を通知したアプリケーションのうち、1つ以上のアプリケーションから、オフライン状態に移行できないことを通知されても、オフライン移行要求を通知した全てのアプリケーションからの応答を待ったのちに、OUS43に対し、アプリケーションがオフライン状態に移行しなかったことを通知する。

**【0168】**

次に、上記処理とは別に、1つのアプリケーションからオフライン状態に移行できないことを通知されると、即座にOUS43にアプリケーションがオフライン状態に移行しなかったことを通知する処理を、図25のシーケンス図を用いて説明する。

**【0169】**

ステップS1001で、NCS31は、プログラム更新開始をネットワークから通知される。ステップS1002で、NCS31は、OUS43に、プログラム更新開始を通知する。通知を受けたOUS43は、ステップS1003で、SCS38に対し、アプリオフライン要求を通知する。

**【0170】**

オフライン要求をOUS43から通知されたSCS38は、アプリA81と、アプリB82に、ステップS1004とステップS1005でアプリオフライン移行要求を通知する。

**【0171】**

アプリオフライン移行要求を通知されたアプリB82は、オフライン状態に移行することができないため、ステップS1006で、SCS38にアプリオフライン応答（NG）を通知する。

**【0172】**

アプリB82からオフライン状態に移行できないことを通知されたSCS38は、アプリA81と、アプリB82に、ステップS1007とステップS1008でアプリオフライン解除要求を通知する。

**【0173】**

そして、SCS38は、OUS43に対し、ステップS1009でアプリケーションがオフラインに移行できなかったことを示すアプリオフライン確定（オフライン移行NG）を通知する。

**【0174】**

このように、SCS38は、オフライン移行要求を通知したアプリケーションのうち、1つのアプリケーションからオフライン状態に移行できないことを通知されると、オフライン移行要求を通知した他のアプリケーションからの応答を待たずに、OUS43に対し、アプリケーションがオフライン状態に移行しなかったことを通知する。

**【0175】**

次に、SCS38が、アプリケーションに対してオフライン移行要求を通知してから時間を計測し、所定の時間が経過すると、応答のあるなしに関わらず、OUS43にアプリオフライン確定を通知する処理について、図26のシーケンス図を用いて説明する。

**【0176】**

ステップS1101で、NCS31は、プログラム更新開始をネットワークから通知され、ステップS1102で、OUS43に、プログラム更新開始を通知する。通知を受けたOUS43は、ステップS1103で、SCS38に対し、アプリオフライン要求を通知する。

【0177】

オフライン要求をOUS43から通知されたSCS38は、アプリA81と、アプリB82に、ステップS1104とステップS1105でアプリオフライン移行要求を通知する。

【0178】

アプリオフライン移行要求を通知されたアプリB82は、オフライン状態に移行することと可能であるので、ステップS1106で、SCS38にアプリオフライン応答(OK)を通知する。

【0179】

その後、所定の時間が経過したため、SCS38は、応答待ちのタイムアウトとし、アプリA81からの応答を待たずに、SCS38は、ステップS1107で、OUS43にアプリオフライン確定を通知する。

【0180】

オフライン確定を通知されたOUS43は、ステップS1108でSCS38に、オフライン状態に移行したアプリケーションが使用している変動エリアを解放させるためにメモリ解放要求を通知する。

【0181】

メモリ解放要求を通知されたSCS38は、アプリA81と、アプリB82に、ステップS1109とステップS1110でメモリ解放要求を通知する。メモリ解放要求を通知されたアプリA81、アプリB82は、それぞれメモリ領域を解放し、ステップS1111、ステップS1112で、SCS38にメモリ解放完了応答を通知する。

【0182】

メモリ解放完了応答を通知されたSCS38は、OUS43に、ステップS1113でメモリ領域の解放が完了したことを示すメモリ解放完了応答を通知する。

【0183】

メモリ解放完了応答を通知されたOUS43は、更新プログラムをダウンロードし、メモリ領域の解放により得られたメモリ領域に更新プログラムを展開する。

【0184】

そして、OUS43は、ステップS1114で、リモートROM更新開始要求をSCS38に通知する。リモートROM更新開始要求を通知されたSCS38は、キーイベントのマスクを行う。

【0185】

次に、SCS38は、プログラムの更新処理を行うために、ステップS1115で、ダウンロードした更新プログラムをエンジンファーム55に送信する。エンジンファーム55は、プログラムの更新を行い、その結果をステップS1116でSCS38に通知する。

【0186】

SCS38は、プログラムを更新した結果が通知されると、ステップS1117で、エンジンファーム55に対し、リブート準備要求を通知する。そして、ステップS1118で、エンジンファーム55からリブート準備完了を通知されると、SCS38は、強制リブートを行い、プログラム更新処理は終了する。

【0187】

このように、SCS38は、アプリケーションに対し、オフライン移行要求を通知するとともに、時間を計測し、オフライン移行要求に対するアプリケーションの応答が、所定の時間が経過しても通知されない場合、OUS43に対し、アプリケーションがオフライン状態に移行したことを通知する。

**【0188】**

次に、上述したタイムアウト処理と、アプリケーションから全ての応答を待ったのちに OUS 43 に通知する処理とを行う SCS 38 の処理について、図 27 を用いて説明する。

**【0189】**

ステップ S1201 で、SCS 38 は、各アプリケーションにアプリオフライン移行要求を通知する。次に、SCS 38 は、ステップ S1202 でタイムアウト時間が経過したかどうか判断する。

**【0190】**

タイムアウト時間が経過している場合は、後述するステップ S1207 へ処理が進む。タイムアウト時間が経過していない場合、SCS 38 は、ステップ S1203 へ処理を進める。ステップ S1203 で、SCS 38 は、任意のアプリケーションからアプリオフライン応答を受信すると、ステップ S1204 で、アプリオフライン応答の受信結果を判断する。

**【0191】**

受信結果がアプリオフライン応答 (NG) の場合、SCS 38 は、アプリオフライン応答 (NG) が発生したことを記憶しておき、ステップ S1206 へ処理を進める。

**【0192】**

また、受信結果がオフライン応答 (OK) の場合、SCS 38 は、ステップ S1206 へ処理を進める。

**【0193】**

ステップ S1206 は、全てのアプリケーションからアプリオフライン応答を受信したかどうかを SCS 38 が判断する処理である。全てのアプリケーションからアプリオフライン応答を SCS 38 が受信していない場合、SCS 38 は、再びステップ S1202 の処理を行う。

**【0194】**

全てのアプリケーションからアプリオフライン応答を SCS 38 が受信した場合、SCS 38 は、ステップ S1207 の処理を行う。

**【0195】**

ステップ S1207 の処理は、SCS 38 が、ここまでの処理でアプリオフライン応答 (NG) の通知があったかどうか判断する処理である。アプリオフライン応答 (NG) の通知がなかった場合、SCS 38 は、ステップ S1208 でアプリケーションがオフライン状態に移行したことを OUS 43 に通知し処理を終了する。

**【0196】**

また、アプリオフライン応答 (NG) の通知があった場合、SCS 38 は、ステップ S1209 でアプリケーションがオフライン状態に移行しなかったことを OUS 43 に通知し処理を終了する。

**【0197】**

次に、タイムアウト処理と、1つのアプリケーションからオフライン状態に移行できないことを通知されると、即座に OUS 43 に通知する処理とを行う SCS 38 の処理について、図 28 を用いて説明する。

**【0198】**

ステップ S1301 で、SCS 38 は、アプリケーションに対してアプリオフライン移行要求を行う。次に、SCS 38 は、ステップ S1302 でタイムアウト時間が経過したかどうか判断する。

**【0199】**

タイムアウト時間が経過している場合は、後述するステップ S1306 へ処理が進む。タイムアウト時間が経過していない場合、SCS 38 は、ステップ S1303 へ処理を進める。ステップ S1303 で、SCS 38 は、任意のアプリケーションからアプリオフライン応答を受信すると、ステップ S1304 で、アプリオフライン応答の受信結果を判断



する。

【0200】

受信結果がアプリオフライン応答 (NG) の場合、SCS 38は、ステップS 1307で、アプリケーションがオフライン状態に移行しなかったことをOUS 43に即座に通知し処理を終了する。

【0201】

また、受信結果がアプリオフライン応答 (OK) の場合、SCS 38は、ステップS 1305へ処理を進める。

【0202】

ステップS 1205は、全てのアプリケーションからオフライン応答を受信したかどうかをSCS 38が判断する処理である。SCS 38が、全てのアプリケーションからオフライン応答を受信していない場合、SCS 38は、再びステップS 1302の処理を行う。

【0203】

SCS 38が、全てのアプリケーションからオフライン応答を受信した場合、SCS 38は、ステップS 1306で、アプリケーションがオフライン状態に移行したことをOUS 43に通知し処理を終了する。

【0204】

次に、IMH 41が、アプリケーションが使用する変動エリアを解放する処理を説明する。

【0205】

まず、図29のシーケンス図を用いて、IMH 41が、アプリケーションが使用する変動エリアを解放する場合の処理を説明する。

【0206】

図29に示されるシーケンス図は、図14のシーケンス図において、IMH 41の処理により変わる部分であるステップS 109からステップS 114までの間の処理を示したものである。

【0207】

図29のステップS 109は、図14と同様に、OUS 43が、SCS 38にメモリ解放要求を通知する。メモリ解放要求を通知されたSCS 38は、図14と異なり、IMH 41にステップS 11でメモリ解放要求を通知する。メモリ解放要求を通知されたIMH 41は、各アプリケーションの変動エリアを解放し、ステップS 12でSCS 38に、メモリ解放完了応答を通知する。IMH 41からメモリ解放完了応答を通知されたSCS 38は、図14と同様に、ステップS 114で、OUS 43に、メモリ解放完了応答を通知する。

【0208】

次に、この処理を行うIMH 41のフローチャートを、図30を用いて説明する。

【0209】

ステップS 1401で、アプリケーションが起動し、ステップS 1402で、アプリケーションが固定エリア用のメモリを確保する。そのとき、IMH 41は、ステップS 1403で、記憶領域情報である固定エリア情報を保持する。

【0210】

次に、アプリケーションがステップS 1404で変動エリア用のメモリを確保すると、IMH 41は、ステップS 1405で、記憶領域情報である変動エリア情報を保持する。

【0211】

変動エリアと固定エリアからなる記憶領域情報は、図31に示される構造となっている。この図31において、例えば、アプリAの固定エリアの記憶領域情報は、固定エリア先頭アドレス0x100000と、固定エリア末尾アドレス0x1FFFFFFと、変動エリア先頭アドレス0x200000と、固定エリア末尾アドレス0x24FFFFの4つの情報からなる。他のアプリケーションも同様である。なお、上記アドレスは、16進表示となっている。また、固定エリアと

変動エリアが連続する場合、変動エリア末尾アドレスは特に必要ない。さらに、各アプリケーションに割り当てられた領域が連続する場合、固定エリア末尾アドレスは特に必要ない。

#### 【0212】

このようにIMH41は、上記記憶領域情報を保持しておくことによって、保持した記憶領域情報に基づき、アプリケーションが使用するメモリ領域を解放することができる。

#### 【0213】

フローチャートの説明に戻り、IMH41は、ステップS1406で、SCS38からのメモリ解放要求を受信すると、ステップS1407で、IMH41は、各アプリケーションの変動エリアを解放する。そして、ステップS1408で、IMH41は、SCS38にメモリ解放完了応答を通知する。

#### 【0214】

このように、変動エリアを解放した状態で、IMH41が、ステップS1409で、SCS38から、オンライン解除を受信すると、IMH41は、ステップS1410で、各アプリケーションの変動エリア情報を再び保持する。

#### 【0215】

以上が、プロセスをオフライン状態して変動エリアを解放する場合の処理についての説明である。次に、プロセスを消滅する場合の処理について説明する。

#### 【0216】

図32は、融合機1が2つのアプリケーション（アプリA81、アプリB82）を消滅して更新プログラムをダウンロードし、フラッシュメモリ69に格納されたプログラムを更新する処理を示すシーケンス図である。

#### 【0217】

以下、シーケンス図の説明をする。ステップS1501で、NCS31は、プログラム更新開始をネットワークから通知され、ステップS1502で、OUS43に、プログラム更新開始を通知する。プログラム更新開始を通知されたOUS43は、ステップS1503で、SCS38にアプリオフライン要求を通知する。なお、このシーケンス図でのオフライン要求は、オフライン状態にしたのちに、さらにアプリケーションを消滅させる要求も含んでいる。

#### 【0218】

アプリオフライン要求をOUS43から通知されたSCS38は、アプリA81と、アプリB82にステップS1504とステップS1505でアプリオフライン移行要求を通知する。

#### 【0219】

アプリオフライン移行要求を通知されたアプリA81、アプリB82は、それぞれオフライン状態に移行することが可能である場合、SCS38に対し、アプリオフライン応答（OK）を通知する。このシーケンス図の場合、アプリA81、アプリB82ともにオフライン状態に移行することが可能であるため、アプリA81、アプリB82は、ステップS1506、ステップS1507で、それぞれアプリオフライン応答（OK）をSCS38に通知している。

#### 【0220】

アプリオフライン応答（OK）を通知されたSCS38は、アプリA81、アプリB82の消滅を実行する。この処理は、プロセス消滅段階に対応する。

#### 【0221】

アプリケーションの消滅を実行したSCS38は、アプリケーションの消滅が完了したことを、ステップS1508で、アプリオフライン確定（プロセスKILL完了）としてOUS43に通知する。アプリオフライン確定を通知されたOUS43は、更新プログラムをダウンロードし、アプリケーションの消滅により得られたメモリ領域に更新プログラムを展開する。

#### 【0222】

次に、OUS 43は、ステップS 1509で、プログラム更新開始要求であるリモートROM更新開始要求をSCS 38に通知する。リモートROM更新開始要求を通知されたSCS 38は、キーイベントのマスクを行う。

【0223】

次に、SCS 38は、プログラムの更新処理を行うために、ステップS 1510で、ダウンロードした更新プログラムをエンジンファーム55に送信する。エンジンファーム55は、プログラムの更新を行い、その結果をステップS 1511でSCS 38に通知する。

【0224】

SCS 38は、プログラムを更新した結果が通知されると、ステップS 1512で、エンジンファーム55に対し、リブート準備要求を行う。そして、ステップS 1513で、エンジンファーム55からリブート準備完了を通知されると、SCS 38は、強制リブートを行い、プログラム更新処理を終了する。

【0225】

以上のように、SCS 38は、プロセスに割り当てられたメモリ領域を解放し、OUS 43は、プログラムを更新する更新プログラムを、NCS 31を用いてダウンロードし、SCS 38により解放されたメモリ領域に展開する。

【0226】

次に、OUS 43と、SCS 38の処理の詳細を、フローチャートを用いて説明する。

【0227】

最初に、OUS 43の処理について、図33のフローチャートを用いて説明する。ステップS 1601で、OUS 43は、NCS 31からプログラム更新開始を通知される。次のステップS 1602で、OUS 43は、受信したプログラム更新開始通知自体が正常なデータであるかどうかをチェックするためのSUM値のチェックを実行し、プログラム更新開始通知が正常データでない場合は、ステップS 1603で、プログラムの更新処理を終了する。

【0228】

ステップS 1602で、プログラム更新開始通知が正常なデータであると判定された場合、OUS 43は、ステップS 1604で、SCS 38にアプリケーションをオフライン状態にさせ、消滅させるためのアプリオフライン要求を通知する。アプリオフライン要求の応答であるアプリオフライン確定をステップS 1605で受信すると、OUS 43は、ステップS 1606で、アプリオフラインが実行完了したかどうか判断する。このステップS 1606の処理は、アプリオフラインが確定しなかったことを通知された場合に、ステップS 1603で、プログラムの更新処理を終了するための処理である。

【0229】

ステップS 1606で、OUS 43は、アプリオフラインの実行が完了したと判断すると、ステップS 1607で、アプリケーションを消滅することにより確保されたメモリ領域に、更新プログラムを展開することが可能かどうか判断する。

【0230】

更新プログラムを展開することが不可能な場合、OUS 43は、ステップS 1609で、確保されたメモリ領域をクリアする。このクリアとは、メモリ領域をプログラムの更新を実行しない状態に設定することを意味する。そして、OUS 43は、ステップS 1610でSCS 38にプログラム更新開始要求であるリモートROM更新開始要求を通知し、処理を終了する。なお、この場合、メモリ領域がプログラムの更新を実行しない状態に設定されているため、プログラムの更新処理は実行されない。

【0231】

ステップS 1607で、OUS 43は、更新プログラムを展開することが可能であると判断すると、ステップS 1608で、更新プログラムをメモリ領域に展開する。そして、OUS 43は、ステップS 1610でSCS 38にプログラム更新開始要求であるリモートROM更新開始要求を通知し、処理を終了する。

**【0232】**

次に、SCS38の処理について、図34のフローチャートを用いて説明する。ステップS1701で、SCS38は、OUS43からのアプリオフライン要求を受信する。アプリオフライン要求を受信したSCS38は、ステップS1702で、各アプリケーションにアプリオフライン移行要求を通知する。

**【0233】**

次のステップS1703で、SCS38は、全てのアプリケーションからの応答が、アプリオフライン応答（OK）ではなかった場合、ステップS1704で、各アプリケーションに、アプリオフライン解除を通知する。これにより、アプリケーションは、制限されていた処理の実行が可能となる。次に、SCS38は、ステップS1705でアプリケーションがオフラインに移行しなかったことをOUS43に通知し、処理を終了する。

**【0234】**

ステップS1703で、全てのアプリケーションからの応答が、アプリオフライン応答（OK）である場合、SCS38は、ステップS1706で、アプリケーションを消滅させる。そして、ステップS1707で、SCS38は、OUS43にアプリオフライン確定を通知する。

**【0235】**

次に、SCS38は、OUS43からステップS1708で、プログラム更新開始要求であるリモートROM更新開始要求を受信すると、ステップS1709で、オペレーションパネル52上のキー・タッチパネル操作イベントをマスクしてキーイベントのマスクを行う。

**【0236】**

このように、プログラムを更新する際に、オペレーションパネル52を無効化することにより、更新処理を迅速に行うことができるとともに、制限されていない機能は実行することができるため、例えば重要な機能をプログラム更新中にもかかわらず実行することができる。

**【0237】**

次に、SCS38は、ステップS1710で、プログラム更新処理を実行し、融合機1を再起動させるためにステップS1711でリブートを実行し処理を終了する。

次に、上記ステップS1706のプロセスを消滅する処理の詳細を図35のフローチャートを用いて説明する。ステップS1801で、SCS38は、OSの中核部分であるカーネルに用意されたプロセス消滅エラー検知処理をマスクする。そして、SCS38は、ステップS1802でプロセス間通信のエラー処理を停止する。そして、ステップS1803で、SCS38は、アプリケーションのプロセスを消滅する。次に、SCS38は、ステップS1804でSCS38が記憶しているアプリケーションの登録情報を抹消し処理を終了する。

**【0238】**

次に、上記処理とは別に、OUS43がダウンロードする更新プログラムのサイズに応じてアプリケーションを消滅する処理について、シーケンス図とフローチャートを用いて説明する。

**【0239】**

図36は、ダウンロードする更新プログラムのサイズに応じてアプリケーションを消滅する処理を示すシーケンス図である。

**【0240】**

ステップS1901で、NCS31は、プログラム更新開始をネットワークから通知され、ステップS1902で、OUS43にプログラムの更新開始を示すプログラム更新開始を通知する。プログラム更新開始を通知されたOUS43は、ステップS1903で、SCS38にアプリオフライン要求を通知する。

**【0241】**

アプリオフライン要求をOUS43から通知されたSCS38は、アプリA81と、ア

プリ B 8 2 にステップ S 1 9 0 4 とステップ S 1 9 0 5 でアプリオフライン移行要求を通知する。

【0242】

アプリオフライン移行要求を通知されたアプリ A 8 1、アプリ B 8 2 は、それぞれオフライン状態になることが可能である場合、SCS 3 8 に対し、アプリオフライン応答 (OK) を通知する。このシーケンス図の場合、アプリ A 8 1、アプリ B 8 2 とともにオフライン状態になることが可能であるため、アプリ A 8 1、アプリ B 8 2 は、ステップ S 1 9 0 6、ステップ S 1 9 0 7 で、それぞれアプリオフライン応答 (OK) を SCS 3 8 に対し通知している。

【0243】

アプリオフライン応答 (OK) を通知された SCS 3 8 は、ステップ S 1 9 0 8 で、OUS 4 3 にアプリケーションがオフライン状態になったことを示すアプリオフライン確定を通知する。

【0244】

アプリケーションのオフライン確定を通知された OUS 4 3 は、ステップ S 1 9 0 9 で SCS 3 8 に処理の実行を制限されたアプリケーションが使用するメモリ領域を解放させるためにメモリ解放要求を通知する。なお、このメモリ解放要求で解放要求されているメモリ領域は、変動エリアである。

【0245】

メモリ解放要求を通知された SCS 3 8 は、アプリ A 8 1 と、アプリ B 8 2 に対し、ステップ S 1 9 1 0 とステップ S 1 9 1 1 でメモリ解放要求を通知する。メモリ解放要求を通知されたアプリ A 8 1、アプリ B 8 2 は、それぞれメモリ領域を解放し、ステップ S 1 9 1 2、ステップ S 1 9 1 3 で、SCS 3 8 に対してメモリ解放完了応答を通知する。

【0246】

メモリ解放完了応答を通知された SCS 3 8 は、ステップ S 1 9 1 4 で OUS 4 3 にメモリの解放が完了したことを示すメモリ解放完了応答を通知する。

【0247】

メモリ解放完了応答を通知された OUS 4 3 は、メモリの空きをチェックするメモリサイズチェックを行う。そして、この時点では、ダウンロードする更新プログラムを展開する分のメモリの空きがまだ得られていないので、OUS 4 3 は、ステップ S 1 9 1 5 で、SCS 3 8 に対し、アプリケーションを構成するプロセスを消滅させるプロセス KILL 要求を通知する。

【0248】

プロセス KILL 要求を通知された SCS 3 8 は、まずアプリ B 8 2 を消滅する。そして、ステップ S 1 9 1 6 で、SCS 3 8 は、OUS 4 3 に対し、プロセス KILL 完了応答を行う。

【0249】

そこで再び OUS 4 3 は、メモリサイズのチェックを行う。しかし、ダウンロードする更新プログラムを展開する分のメモリの空きがまだ得られていないので、OUS 4 3 は、ステップ S 1 9 1 7 で、SCS 3 8 に再びプロセス KILL 要求を通知する。

【0250】

再びプロセス KILL 要求を通知された SCS 3 8 は、アプリ A 8 1 を消滅する。そして、ステップ S 1 9 1 8 で、SCS 3 8 は、OUS 4 3 にプロセス KILL 完了応答を通知する。

【0251】

プロセス KILL 完了応答を通知された OUS 4 3 は、メモリサイズのチェックを行う。アプリ A 8 1 の消滅により、ダウンロードする更新プログラムを展開する分の空きが得られたので、OUS 4 3 は、更新プログラムをダウンロードし、展開する。

【0252】

そして、OUS 4 3 は、ステップ S 1 9 1 9 で、リモート ROM 更新開始要求を SCS

38に通知する。リモートROM更新開始要求を通知されたSCS38は、キーイベントのマスクを行う。

【0253】

次に、SCS38は、プログラムの更新処理を行うために、ステップS1920で、ダウンロードされた更新プログラムをエンジンファーム55に送信する。エンジンファーム55は、プログラムの更新を行い、その結果をステップS1921でSCS38に通知する。

【0254】

SCS38は、プログラムを更新した結果が通知されると、ステップS1922で、エンジンファーム55に対し、リポート準備要求を行う。そして、ステップS1923で、エンジンファーム55からリポート準備完了を通知されると、SCS38は、強制リポートを行い、プログラム更新処理は終了する。

【0255】

このように、SCS38はダウンロードする更新プログラムのサイズに応じて、プロセスに割り当てられたメモリ領域を解放する。

【0256】

次に、上述した処理におけるOUS43、SCS38の処理の詳細を、フローチャートを用いて説明する。

【0257】

最初に、OUS43の処理について、図37のフローチャートを用いて説明する。ステップS2001で、OUS43は、NCS31からプログラム更新開始通知を受信する。次のステップS2002で、OUS43は、受信したプログラム更新開始通知自体が正常なデータであるかどうかをチェックするためのSUM値のチェックを実行し、プログラム更新開始通知が正常データでない場合は、ステップS2003で、プログラムの更新処理を終了する。

【0258】

ステップS2002で、プログラム更新開始通知が正常なデータであると判断された場合、OUS43は、ステップS2004で、SCS38にアプリオフライン要求を通知する。ステップS2005で、OUS43は、アプリオフライン要求の応答であるアプリオフライン確定を受信すると、ステップS2006で、アプリオフラインが実行完了したかどうか判断する。

【0259】

ステップS2006で、OUS43が、アプリオフラインが確定しなかったと判断した場合、ステップS2003で、プログラムの更新処理を終了する。

【0260】

ステップS2006で、OUS43が、アプリオフラインの実行が完了したと判断した場合、ステップS2007で、SCS38にメモリ解放要求を通知する。そして、ステップS2008で、OUS43は、SCS38からメモリ解放完了を受信するとステップS2009で、更新プログラムの展開に必要なメモリ領域が確保できたかどうか判断する。

【0261】

OUS43が、メモリ領域が確保できなかったと判断した場合、OUS43は、ステップS2010で、規定の優先順にプロセスを消滅することが可能かどうか判断する。

【0262】

なお、規定の優先順とは、予め定められた順番であり、図38に示されるように、例えば融合機1におけるプロセスの優先度に応じて定めておいた「デフォルト」の順番か、プロセスに割り当てられたメモリ領域のサイズである「メモリサイズ」の大きい順番か、「共有メモリ」（図4参照）におけるプロセスに割り当てられたメモリ領域の先頭アドレスの位置で定まる順番などである。

【0263】

このように、プロセスに割り当てられたメモリ領域をSCS38が解放する順番を予め

定めておくことにより、例えば「デフォルト」で定められた使用頻度が多く優先度が高いアプリケーションに割り当てられたメモリ領域は、最後まで解放されないため、プログラム更新中にも、そのアプリケーションを実行することが可能となる。

【0264】

また、SCS38が解放する順番として、アプリケーションを消滅したのちに、アプリケーションが共通で使用する人が多いコントロールサービス層9のモジュールを消滅しても良い。

【0265】

上記いずれか順番でプロセスを消滅することが可能であれば、OUS43は、ステップS2011で、SCS38に対し、プロセスKILL要求を通知する。そして、OUS43は、ステップS2012で、SCS38からプロセスKILL完了を受信すると、再びステップS2009の判断を行う。

【0266】

ステップS2010の処理に戻り、OUS43が、規定の優先順でプロセスを消滅することが不可能であると判断した場合、ステップS2015で、OUS43はリブートを実行するかどうか判断する。このステップS2015の処理は、1回でもアプリケーションを消滅させたかどうかにより、リブートを実行するかどうか判断する処理である。

【0267】

OUS43が、1回もアプリケーションを消滅していなければ、ステップS2017で、OUS43は、SCS38にアプリオフライン解除を通知し、処理を終了する。これにより、アプリケーションは、制限されていた処理の実行が可能な状態に復帰する。

【0268】

OUS43が、1回以上アプリケーションを消滅していれば、ステップS2016で、OUS43は、SCS38にリブート要求を通知し、処理を終了する。

【0269】

ステップS2009の処理に戻り、OUS43が更新プログラムの展開に必要なメモリ領域が確保できた場合、OUS43は、ステップS2013で更新プログラムをメモリ領域に展開する。そして、OUS43は、ステップS2014でSCS38にリモートROM更新開始要求を通知し、処理を終了する。

【0270】

次に、SCS38の処理について、図39のフローチャートを用いて説明する。ステップS2101で、SCS38は、OUS43からアプリオフライン要求を受信する。アプリオフライン要求を受信したSCS38は、ステップS2102で、各アプリケーションにアプリオフライン移行を要求する。

【0271】

次のステップS2103で、SCS38は、全てのアプリケーションからの応答が、アプリオフライン応答（OK）ではなかった場合、ステップS2104で、各アプリケーションに、アプリオフライン解除を通知する。そして、SCS38は、ステップS2105で、OUS43にアプリケーションがオフラインに移行しなかったことを通知し、処理を終了する。

【0272】

ステップS2103で、全てのアプリケーションからの応答が、アプリオフライン応答（OK）である場合、SCS38は、ステップS2106で、OUS43に対し、アプリオフライン確定を通知する。

【0273】

次に、SCS38は、ステップS2107で、OUS43からメモリ解放要求を受信すると、ステップS2108でアプリケーションにメモリの解放処理を実行させ、ステップS2109でOUS43にメモリ解放完了を通知する。

【0274】

SCS38は、OUS43にメモリ解放完了を通知すると、OUS43の要求により処

理が分岐する。まず、SCS38がOUS43からプロセスKILL要求を通知された場合、SCS38は、ステップS2110からステップS2111に分岐し、アプリケーションを消滅する。そして、SCS38はステップS2112でOUS43にプロセスKILL完了を通知し、再びステップS2110の処理を行う。

【0275】

ステップS2110で、SCS38がプロセスKILL要求を通知されなかった場合、SCS38は、ステップS2110からステップS2113に分岐し、OUS43からリモートROM更新開始要求を通知されたかどうか判断する。

【0276】

ステップS2113でSCS38がリモートROM更新開始要求を通知された場合、SCS38は、ステップS2114で、キー・タッチパネル操作イベントをマスクしてキーイベントのマスクを行う。そして、SCS38は、ステップS2115プログラム更新処理を実行する。また、次のステップS2116で、ダウンロードの実行を開始したことやプログラムの更新状況などを、RS42（図1参照）からネットワークを通してサービスセンタなどに通知し、ステップS2117でリブートを行い処理を終了する。

【0277】

なお、サービスセンタとは、例えば融合機1の保守や点検を行うために製造元に備えられたサーバや、更新プログラムの配布元に備えられたサーバなどである。また、各アプリケーションやモジュールの状態情報をサービスセンタに送信してもよいし、サービスセンタではそのログを記録しておいてもよい。

【0278】

ステップS2113で、SCS38がリモートROM更新開始要求を受信しなかった場合、SCS38は、ステップS2118で、OUS43からリブート要求を受信したかどうか判断する。

【0279】

OUS43からの通知がリブート要求ではない場合、SCS38は、再びステップS2110の処理を行う。OUS43からの通知がリブート要求の場合、SCS38は、ステップS2119でダウンロードの実行が失敗し、リブート処理を行うことをRS42からネットワークを通してサービスセンタなどに通知し、ステップS2117でリブートを行い、処理を終了する。

【図面の簡単な説明】

【0280】

【図1】 本発明による融合機の一実施例の構成図である。

【図2】 本発明による融合機の一実施例のハードウェア構成図である。

【図3】 メモリマップを示す図である。

【図4】 共有メモリのメモリマップを示す図である。

【図5】 プロセス稼動状況を示す図である。

【図6】 プロセスのオフライン状況を示す図である。

【図7】 実行を制限された処理を示す図である。

【図8】 解放されたメモリのメモリマップを示す図である。

【図9】 プロセス消滅状況を示す図である。

【図10】 メモリマップを示す図である。

【図11】 エラーハンドラの登録の様子を示す図である。

【図12】 プロセスの消滅によるエラーを示す図である。

【図13】 エラーを回避した様子を示す図である。

【図14】 プログラムを更新する処理を示すシーケンス図である。

【図15】 OUSの処理を示すフローチャートである。

【図16】 SCSの処理を示すフローチャートである。

【図17】 アプリケーションの処理を示すフローチャートである。

【図18】 オフライン状態に移行する処理を示すフローチャートである。



- 【図 19】 オフライン状態における処理を示すフローチャートである。
- 【図 20】 スキャナアプリの内部構造の概略図である。
- 【図 21】 スキャナアプリのオフライン処理イベントシーケンス図である。
- 【図 22】 スキャナアプリのオフライン解除イベントシーケンス図である。
- 【図 23】 スキャナアプリのオフライン状態説明図である。
- 【図 24】 オフライン NG 時の処理を示すシーケンス図である。
- 【図 25】 オフライン NG 時の処理を示すシーケンス図である。
- 【図 26】 タイムアウト処理を示すシーケンス図である。
- 【図 27】 SCS の処理を示すフローチャートである。
- 【図 28】 SCS の処理を示すフローチャートである。
- 【図 29】 IMH の処理を示すシーケンス図である。
- 【図 30】 IMH の処理を示すフローチャートである。
- 【図 31】 記憶領域情報を示す図である。
- 【図 32】 プログラムを更新する処理を示すシーケンス図である。
- 【図 33】 OUS の処理を示すフローチャートである。
- 【図 34】 SCS の処理を示すフローチャートである。
- 【図 35】 プロセスの消滅の処理を示すフローチャートである。
- 【図 36】 プログラムを更新する処理を示すシーケンス図である。
- 【図 37】 OUS の処理を示すフローチャートである。
- 【図 38】 消滅順番を示す図である。
- 【図 39】 SCS の処理を示すフローチャートである。

【符号の説明】

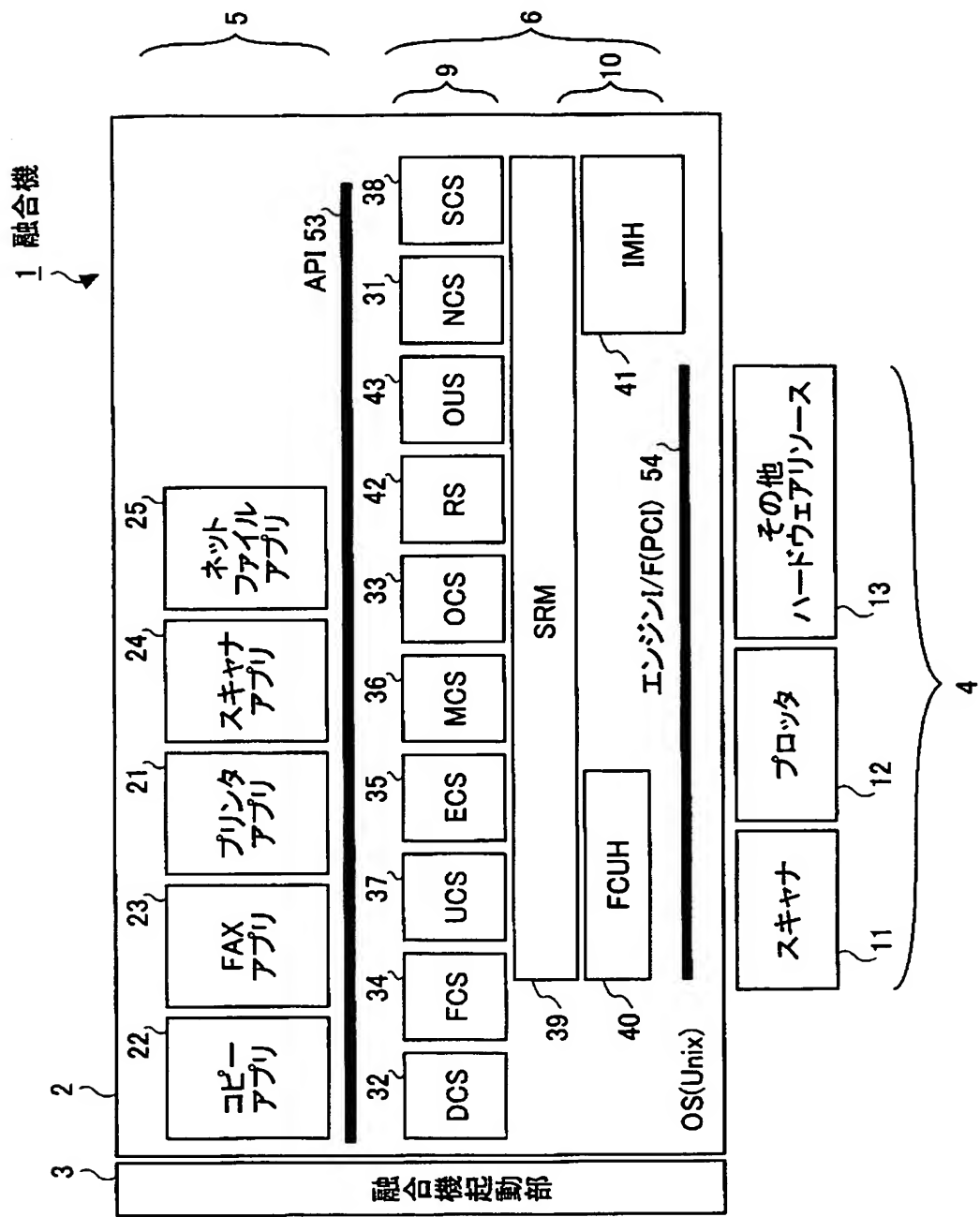
【0281】

- 1 融合機
- 2 ソフトウェア群
- 3 融合機起動部
- 4 ハードウェア資源
- 5 アプリケーション層
- 6 プラットホーム層
- 9 コントロールサービス層
- 10 ハンドラ層
- 11 スキャナ
- 12 プロッタ
- 13 その他のハードウェアリソース
- 21 プリンタアプリ
- 22 コピーアプリ
- 23 ファックスアプリ
- 24 スキャナアプリ
- 25 ネットファイルアプリ
- 31 ネットワークコントロールサービス (NCS)
- 32 デリバリーコントロールサービス (DCS)
- 33 オペレーションパネルコントロールサービス (OCS)
- 34 ファックスコントロールサービス (FCS)
- 35 エンジンコントロールサービス (ECS)
- 36 メモリコントロールサービス (MCS)
- 37 ユーザインフォメーションコントロールサービス (UCS)
- 38 システムコントロールサービス (SCS)
- 39 システムリソースマネージャ (SRM)
- 40 ファックスコントロールユニットハンドラ (FCUH)
- 41 イメージメモリハンドラ (IMH)

- 4 2 リモートサービス (R S)
- 4 3 オンデマンドアップデートサービス (O U S)
- 5 2 オペレーションパネル
- 5 3 アプリケーションプログラムインターフェース (A P I)
- 5 4 エンジン I / F
- 5 5 エンジンファーム
- 6 0 コントローラボード
- 6 1 C P U
- 6 2 ノースブリッジ (N B)
- 6 3 システムメモリ (M E M - P)
- 6 4 ローカルメモリ (M E M - C)
- 6 5 ハードディスク装置 (H D D)
- 6 6 A S I C
- 6 7 A G P (Accelerated Graphics  
Port)
- 6 8 ファックスコントロールユニット (F C U)
- 6 9 G 3 対応ユニット
- 7 0 G 4 対応ユニット
- 7 1、7 2、7 3、7 4、7 8 メモリマップ
- 7 5 V M
- 7 9 エンジン
- 8 0 フラッシュメモリ
- 8 1 アプリ A
- 8 2 アプリ B
- 8 3 配信プロセス
- 8 4 T W A I N プロセス
- 8 5 S Y S O P プロセス
- 1 7 3 サウスブリッジ (S B)
- 1 7 4 N I C
- 1 7 5 U S B デバイス
- 1 7 6 I E E E 1 3 9 4 デバイス
- 1 7 7 セントロニクスデバイス
- 2 0 1 S C S プロセス
- 2 0 2 S C S メインタスク
- 2 0 3 アプリエラーハンドル
- 2 0 4 アプリプロセス
- 2 0 5 アプリメインタスク
- 2 0 6 アプリエラーハンドラ

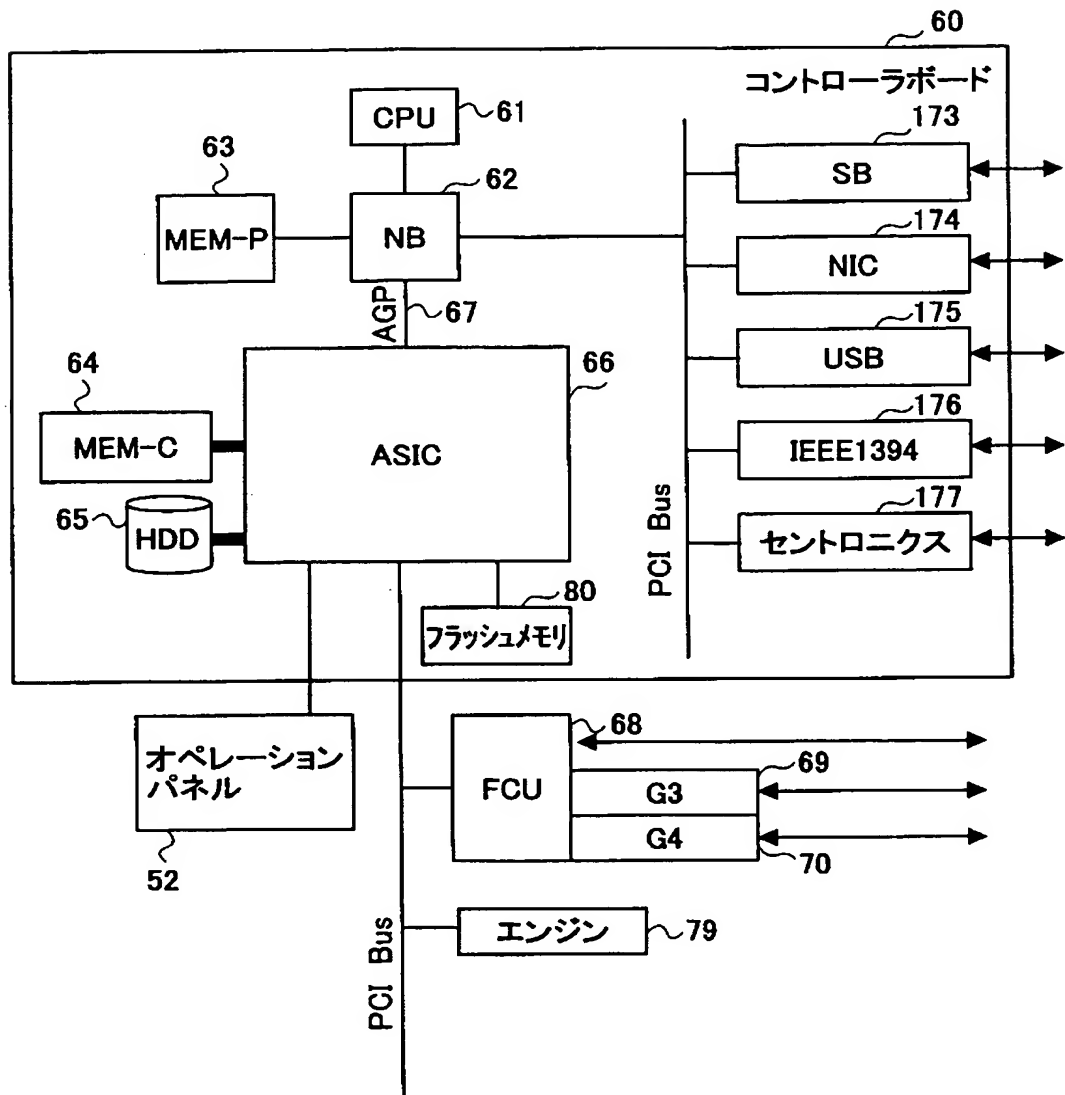
【書類名】 図面  
【図 1】

本発明による融合機の一実施例の構成図



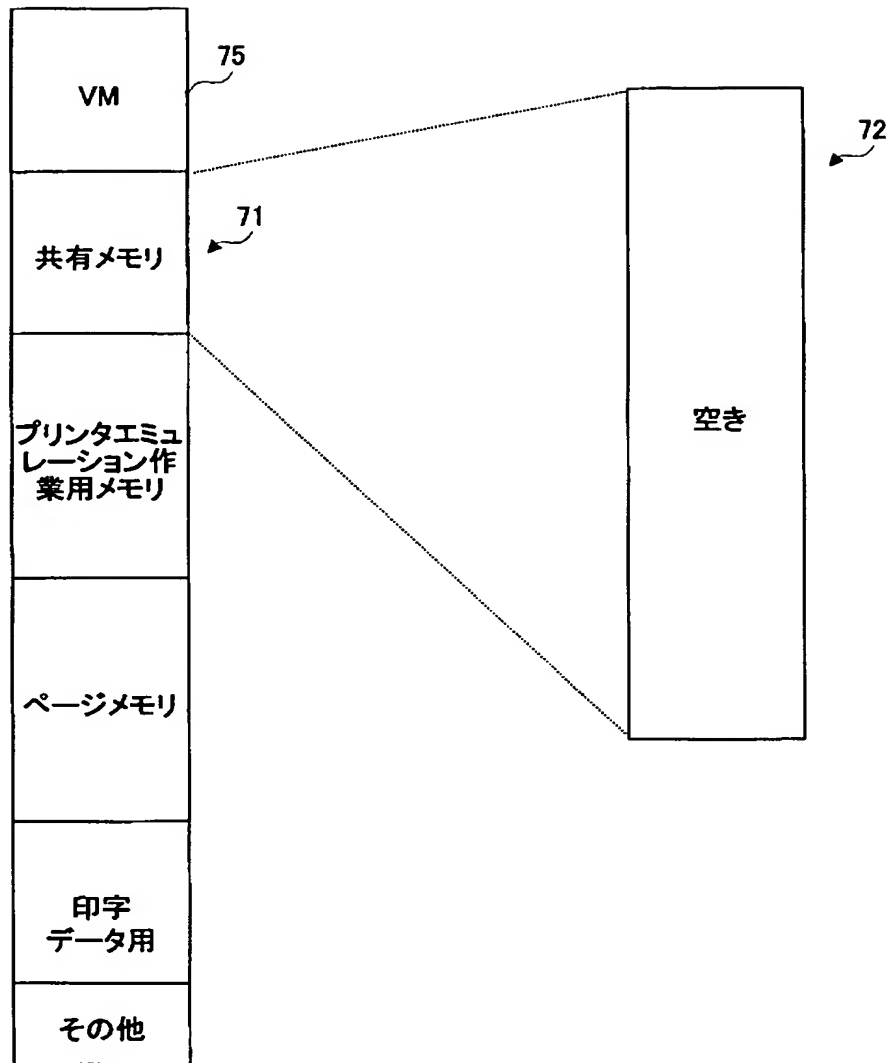
【図 2】

## 本発明による融合機の一実施例のハードウェア構成図



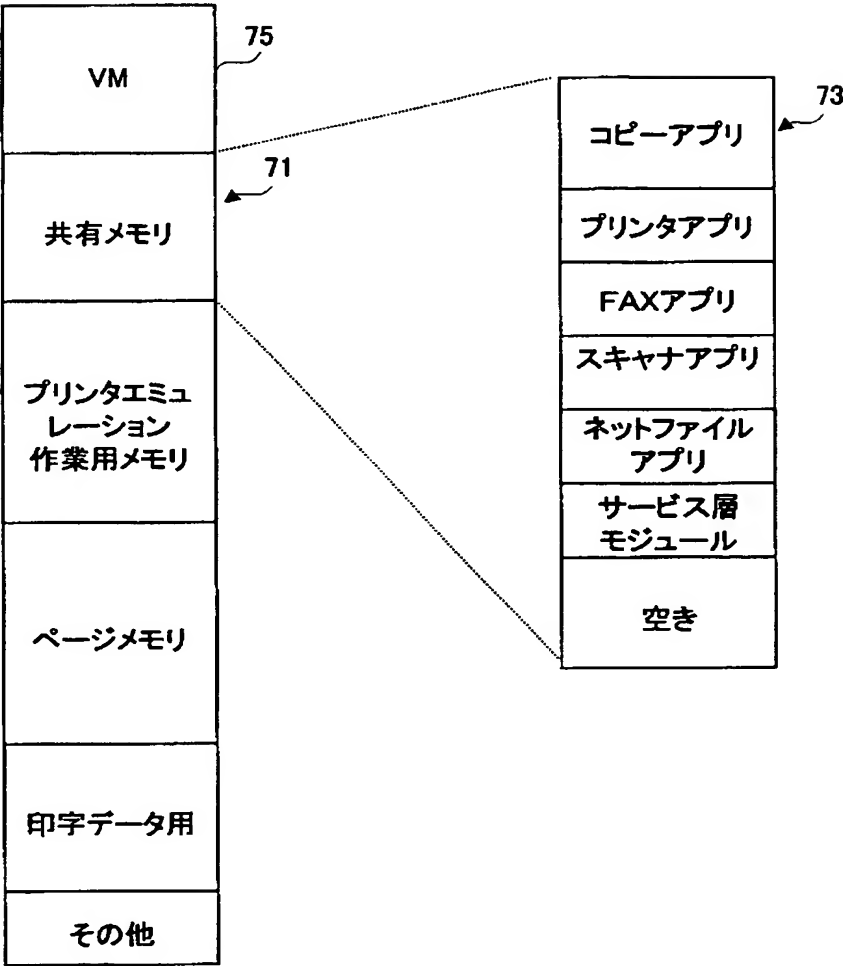
【図 3】

メモリマップを示す図



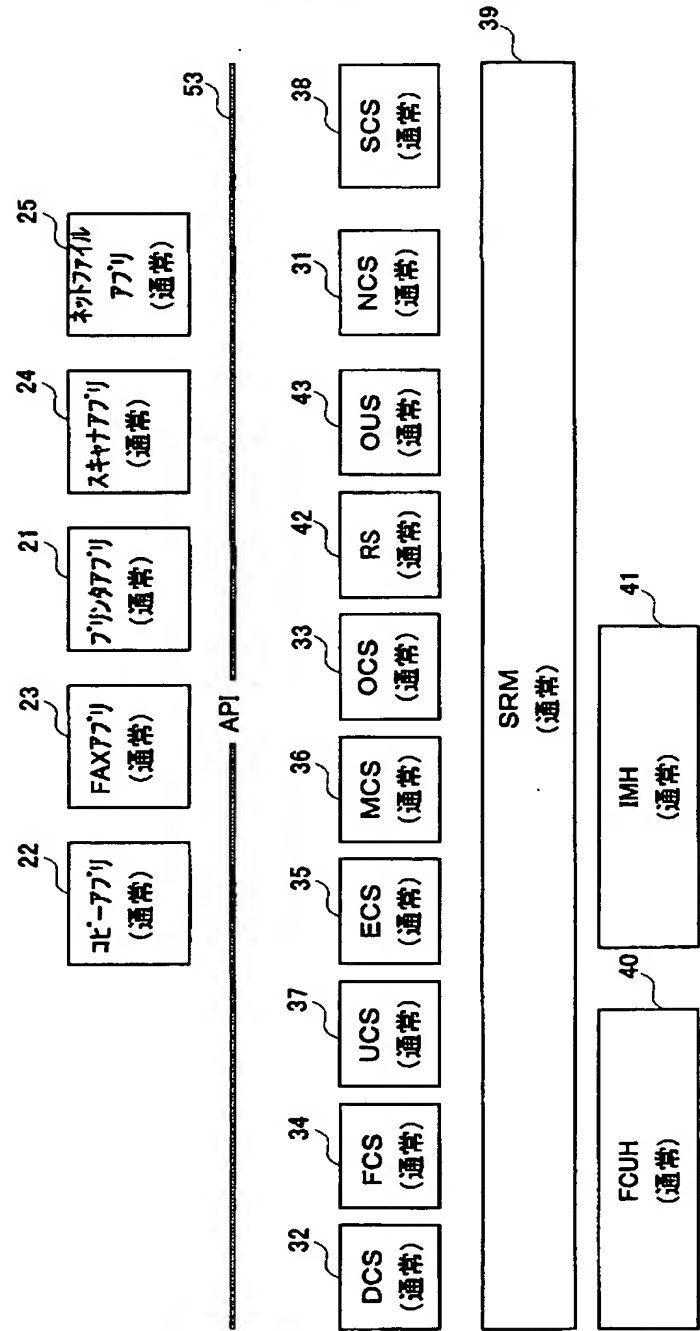
【図 4】

共有メモリのメモリマップを示す図



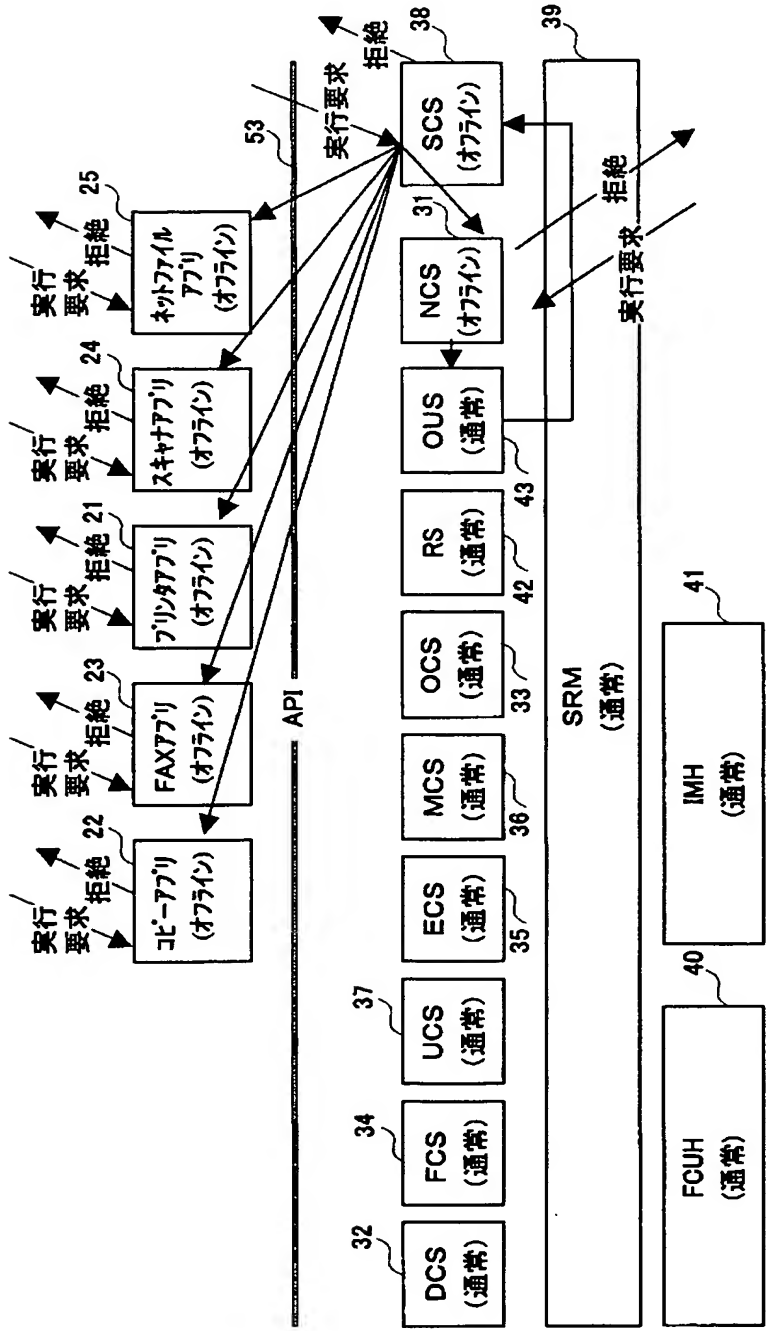
【図 5】

プロセス稼動状況を示す図



【図 6】

プロセスのオフライン状況を示す図





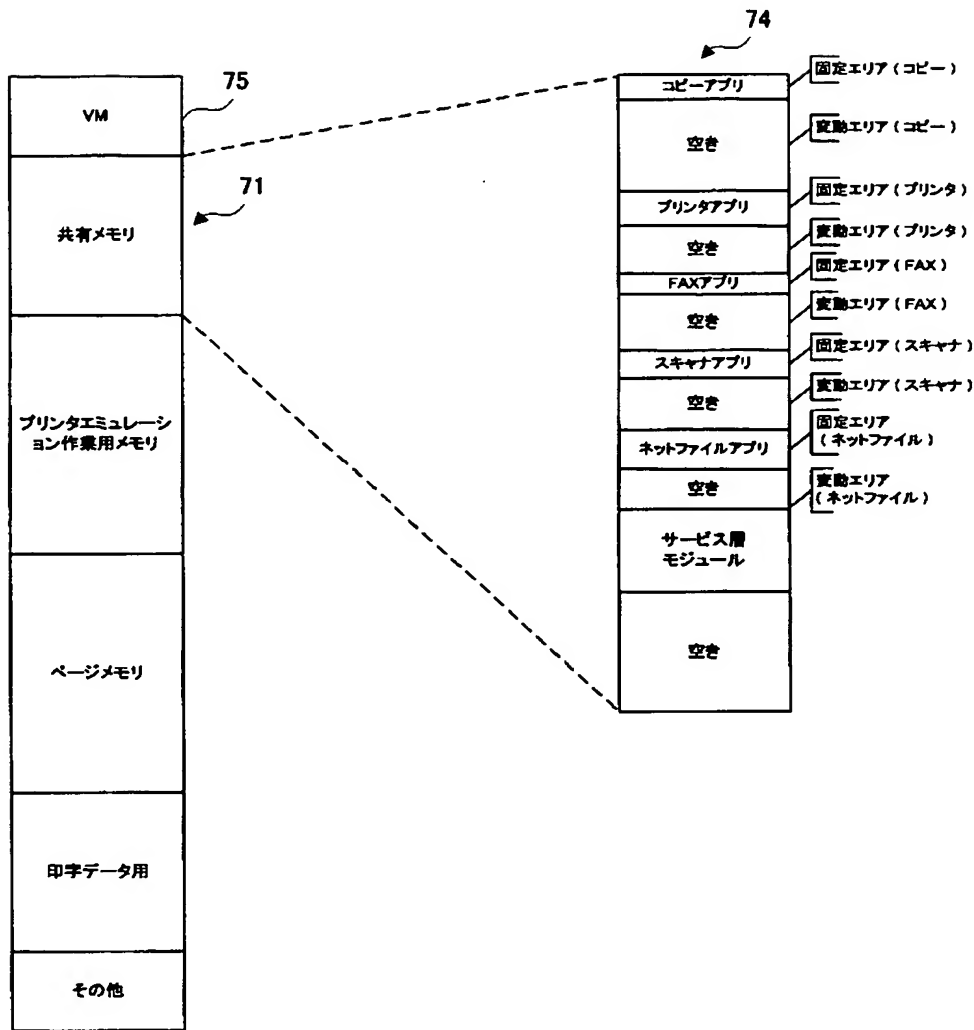
【図 7】

実行を制限された処理を示す図

コピーアプリ	FAXアプリ	プリンタアプリ	スキャナアプリ	ネットファイルアプリ	SCS	NCS
・読み取り ・印刷 ・LS蓄積	・送受信 ・LS蓄積 ・受信文書印刷	・印刷 ・LS蓄積	・読み取り ・LS蓄積	・印刷 ・LS蓄積	・操作部イベントをマスク ・ジョブの実行要求	・通信イベント

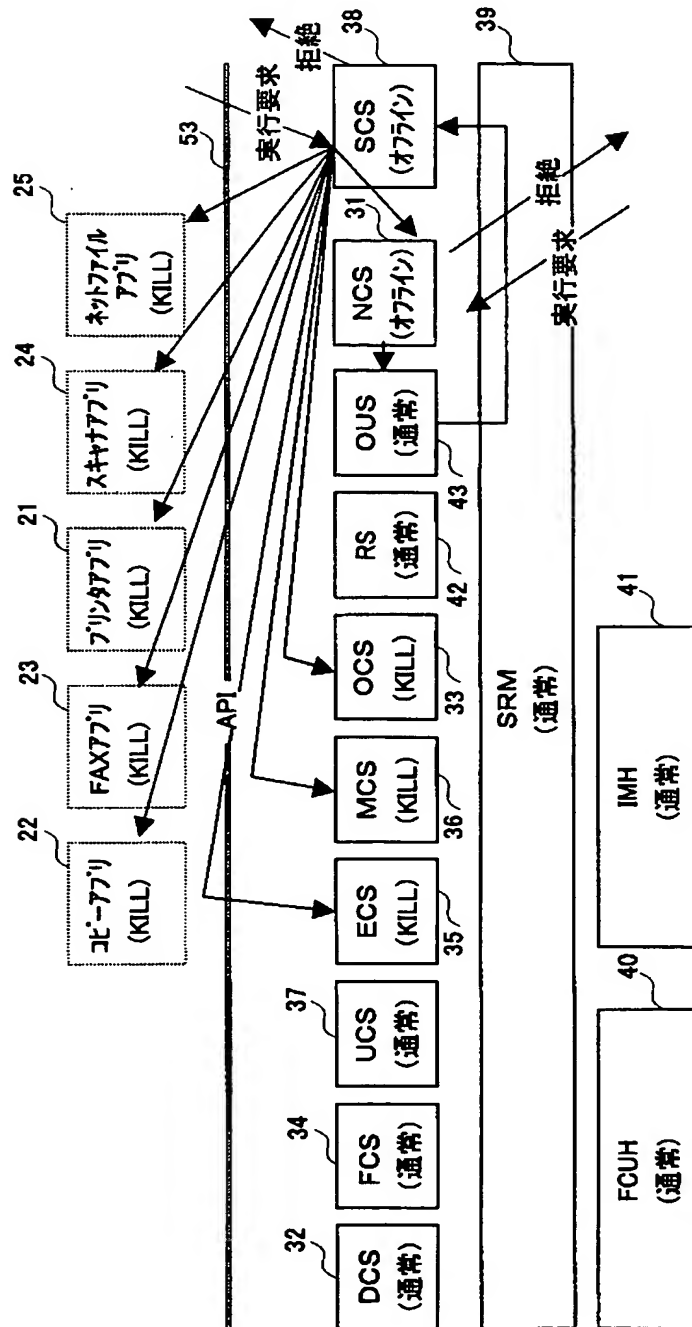
【図 8】

解放されたメモリのメモリマップを示す図



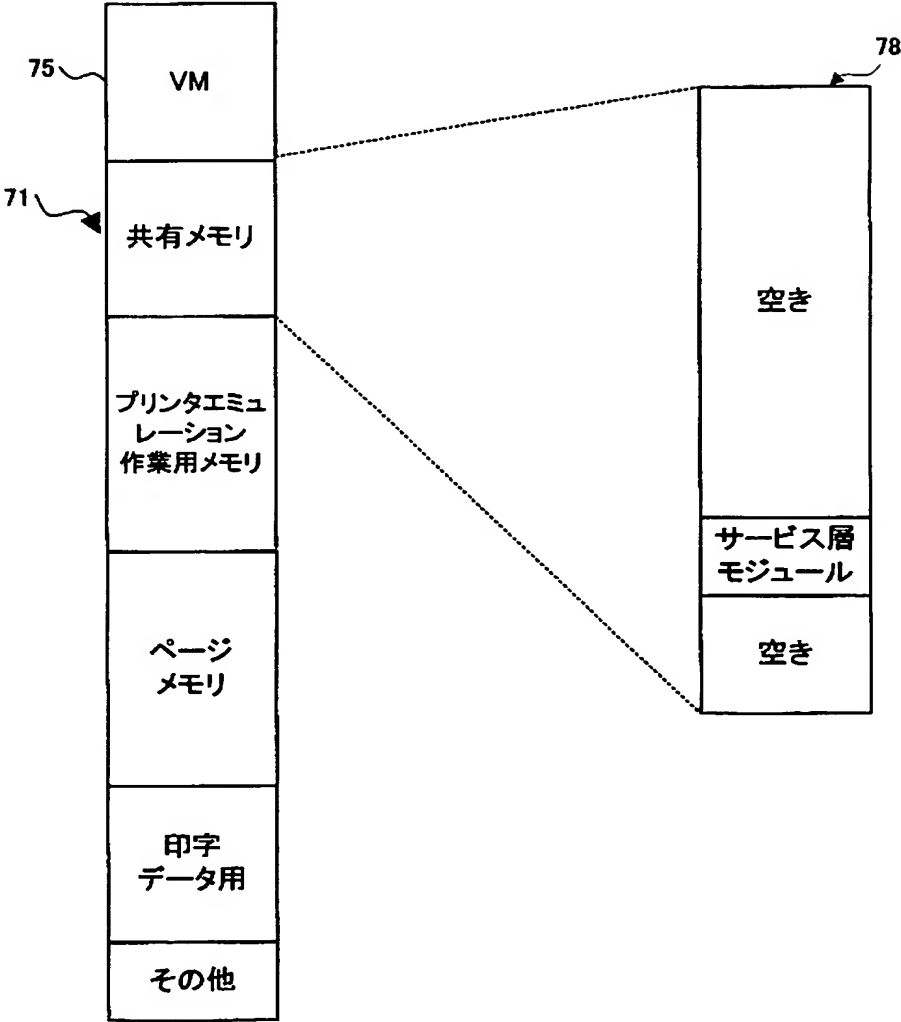
【図9】

プロセス消滅状況を示す図



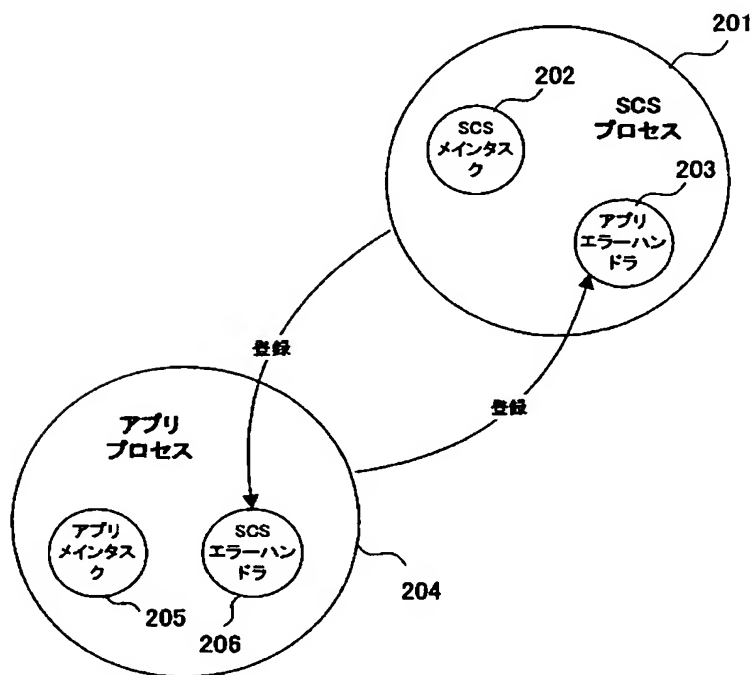
【図 10】

メモリマップを示す図



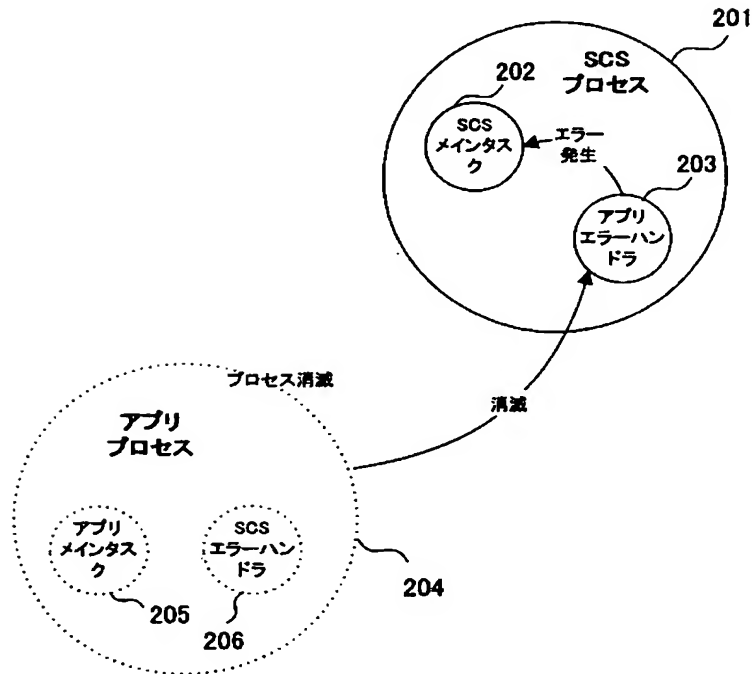
【図 11】

エラーハンドラの登録の様子を示す図



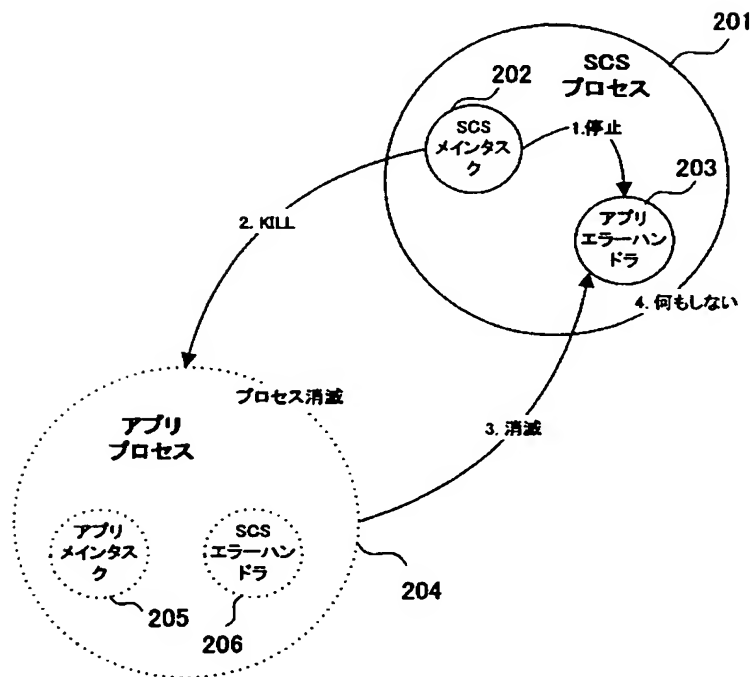
【図 12】

プロセスの消滅によるエラーを示す図



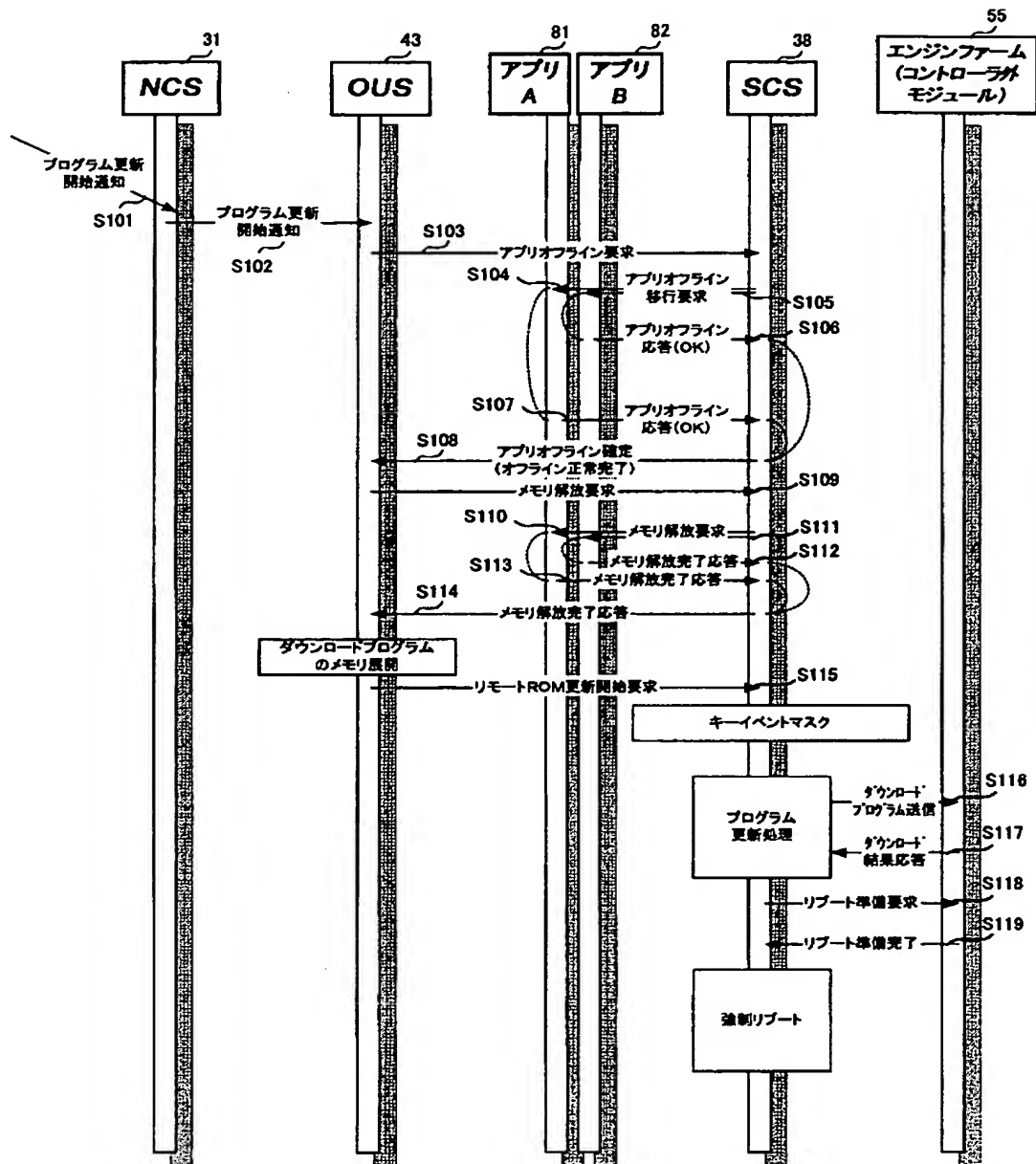
【図 13】

## エラーを回避した様子を示す図



【図 14】

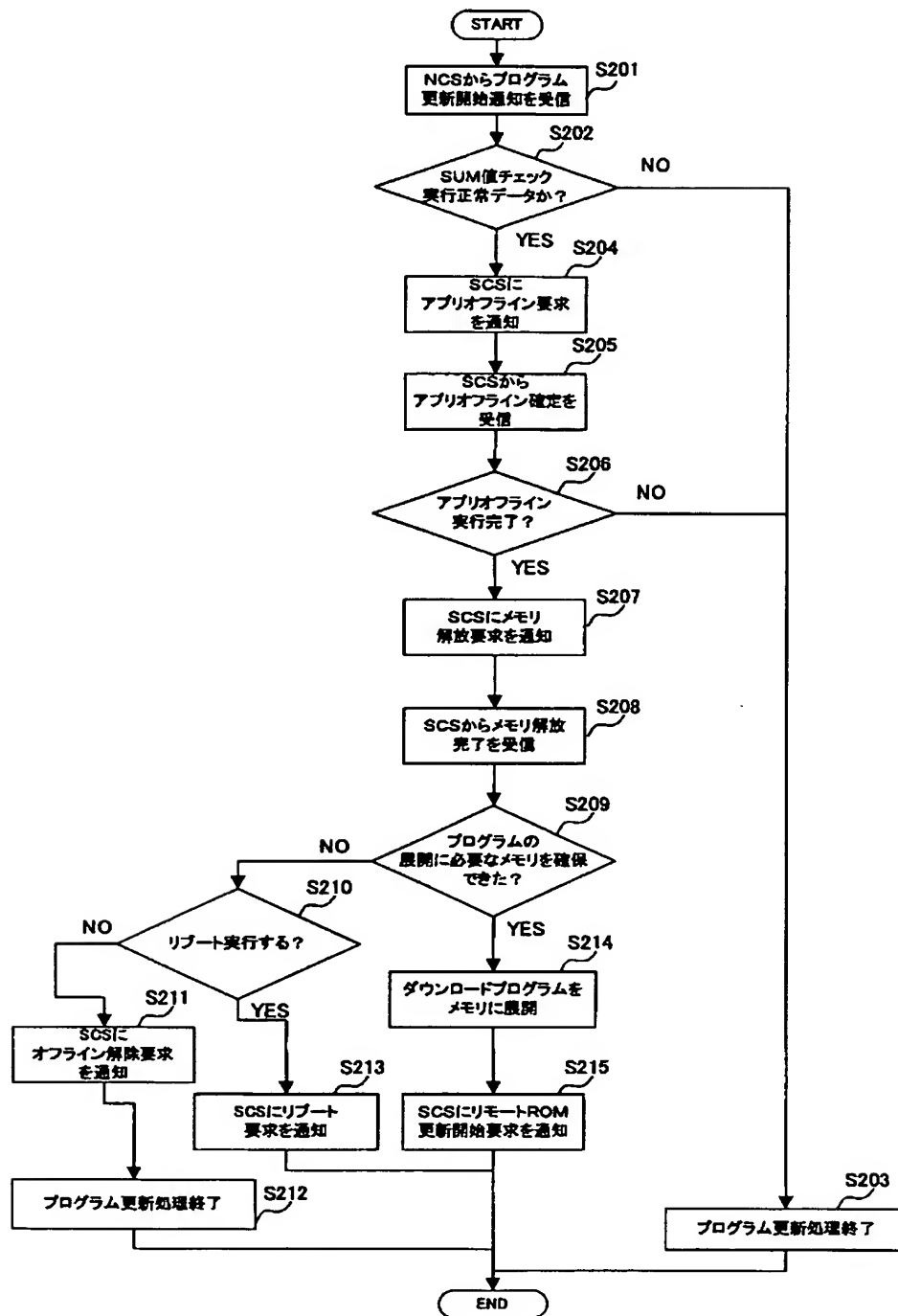
## プログラムを更新する処理を示すシーケンス図





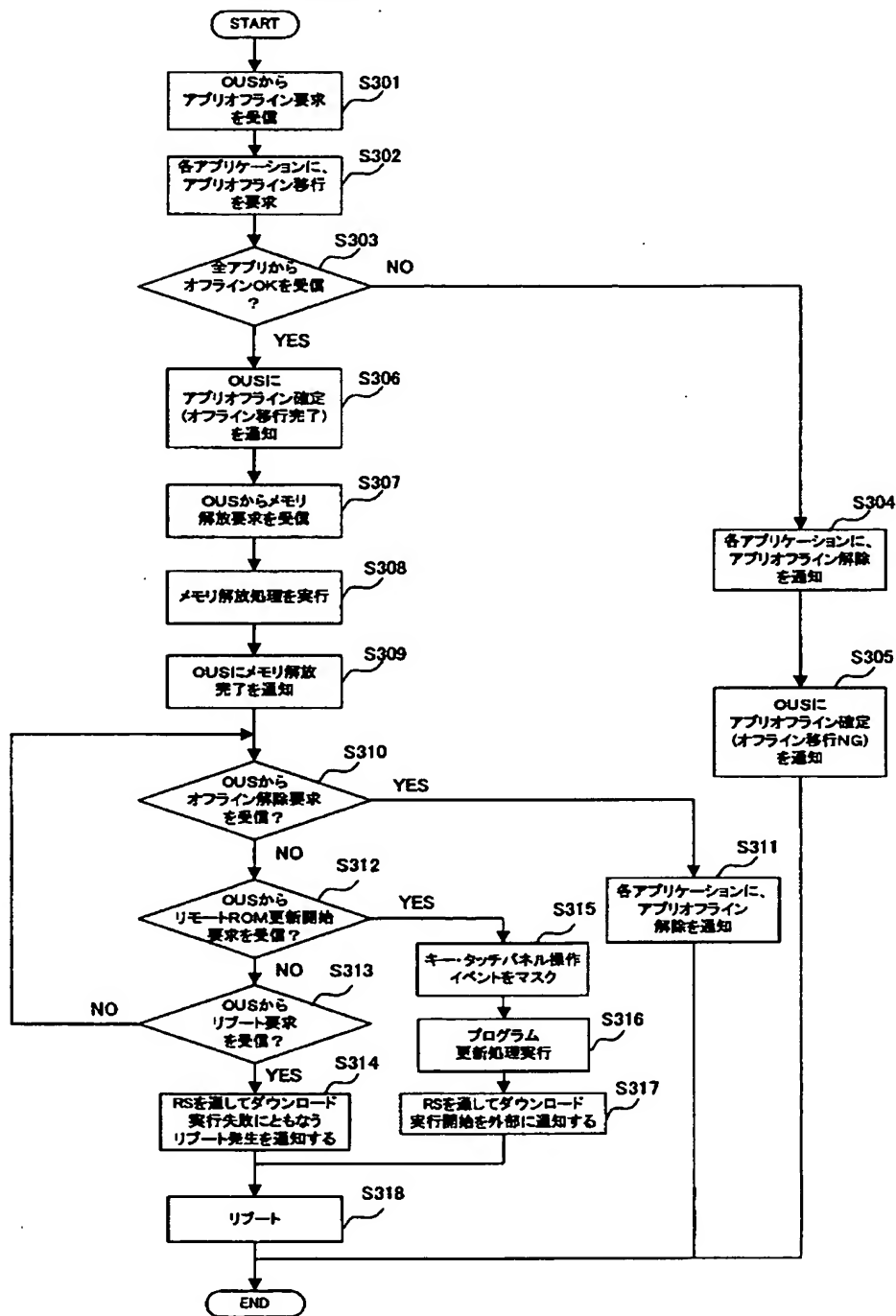
【図 15】

## OUSの処理を示すフローチャート



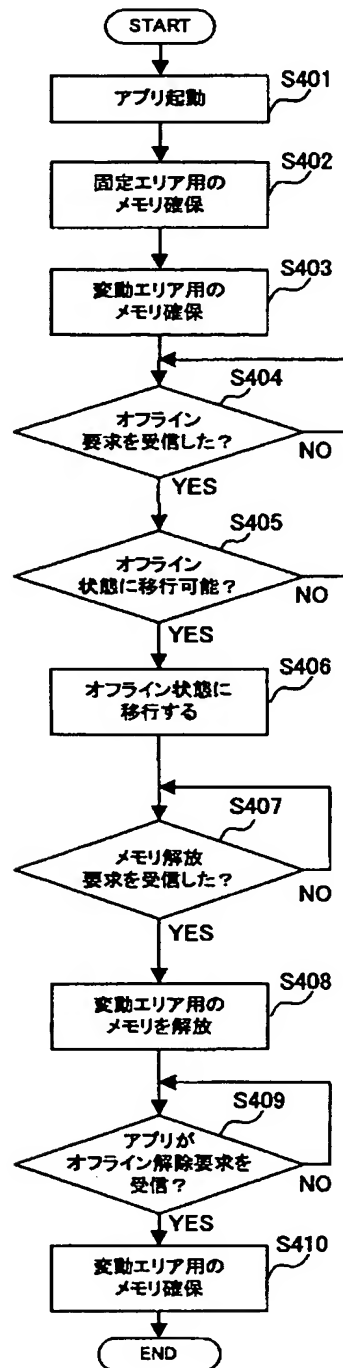
【図16】

## SCSの処理を示すフローチャート



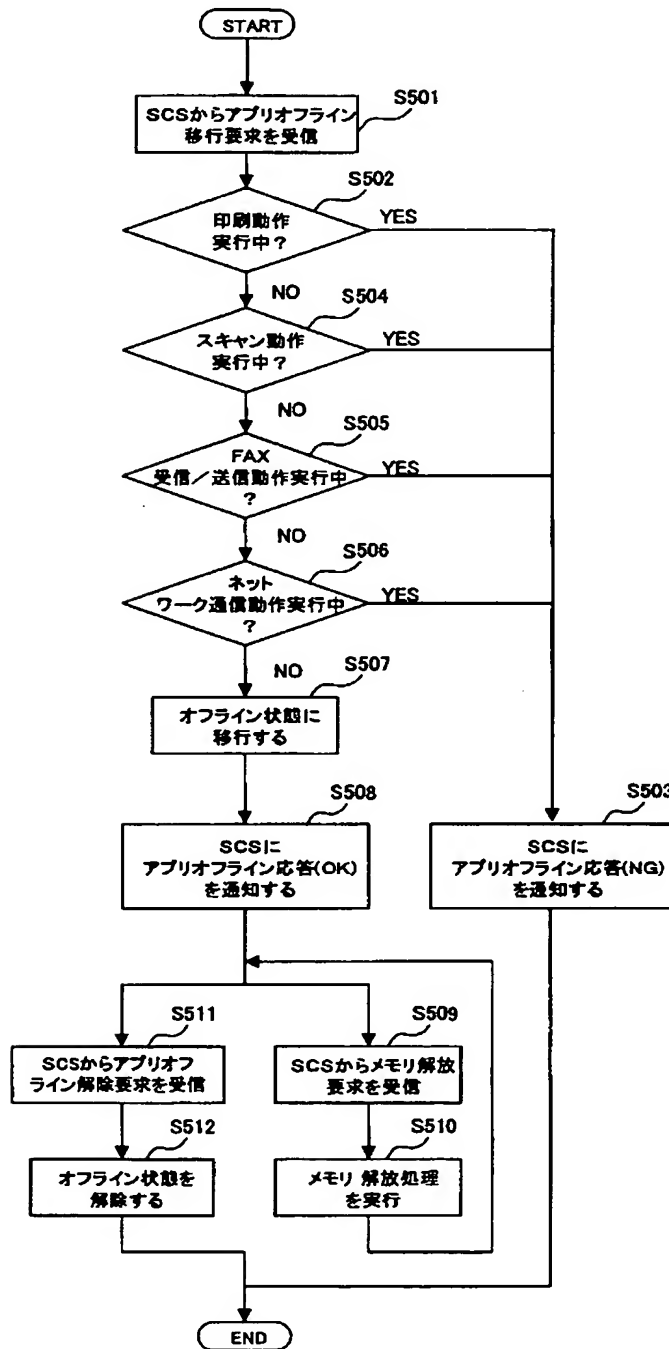
【図 17】

## アプリケーションの処理を示すフローチャート



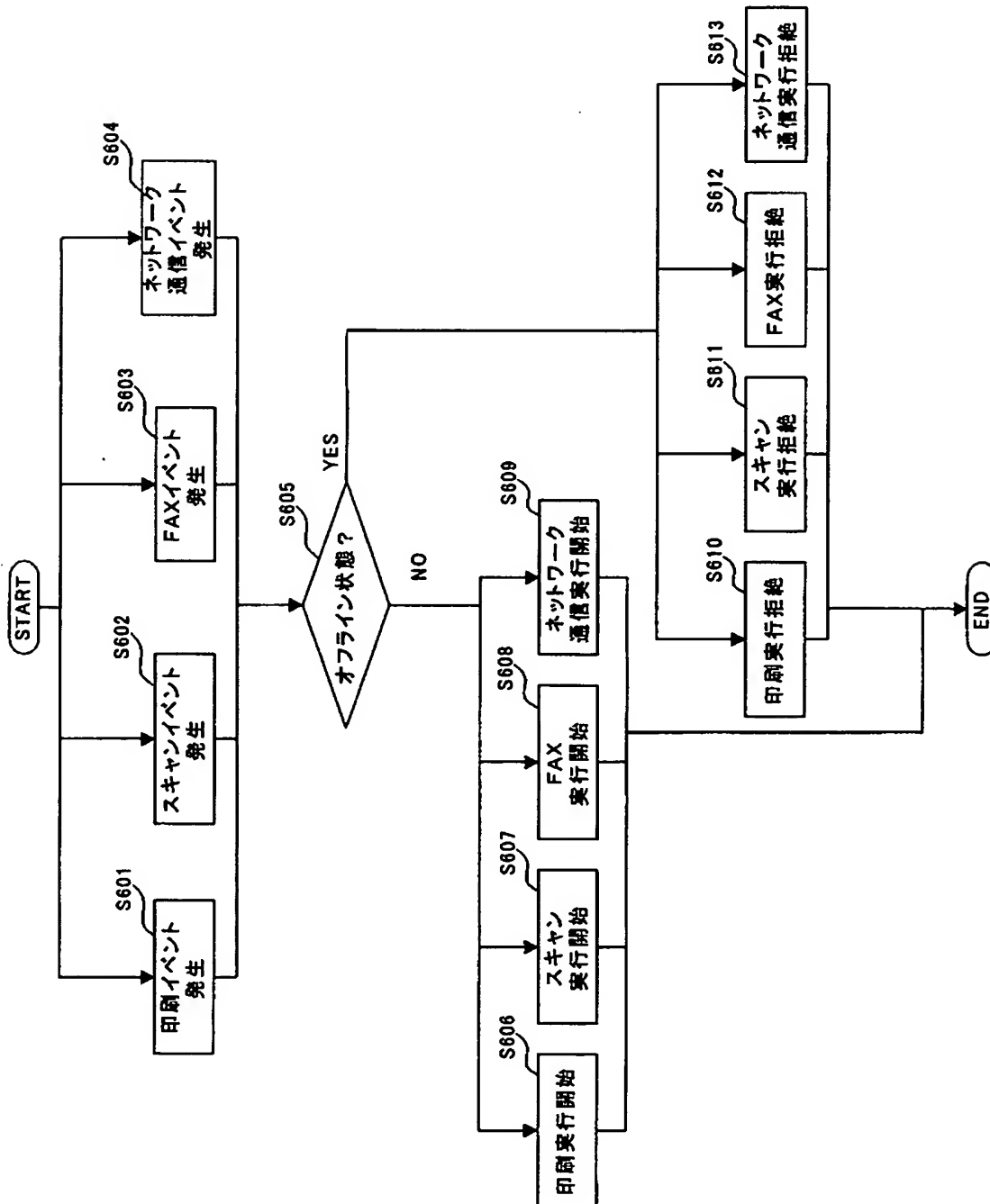
【図 18】

## オフライン状態に移行する処理を示すフローチャート



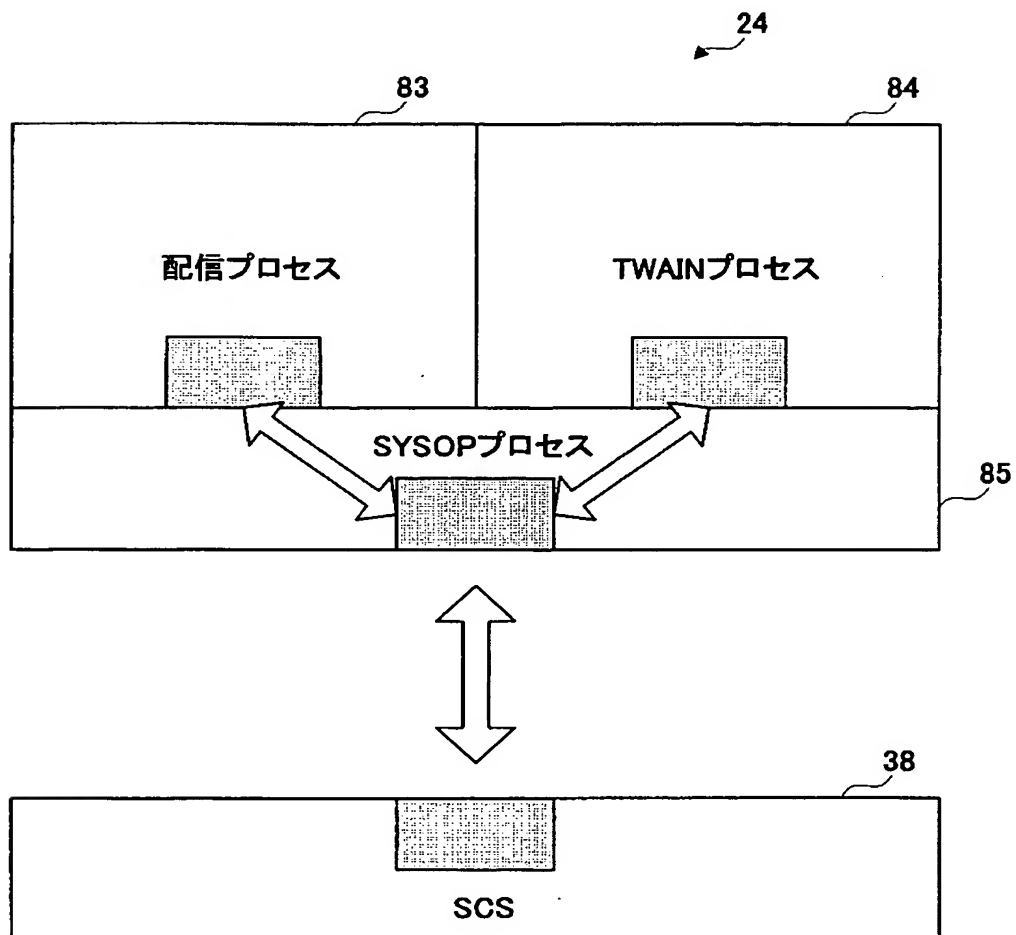
【図 19】

## オフライン状態における処理を示すフローチャート



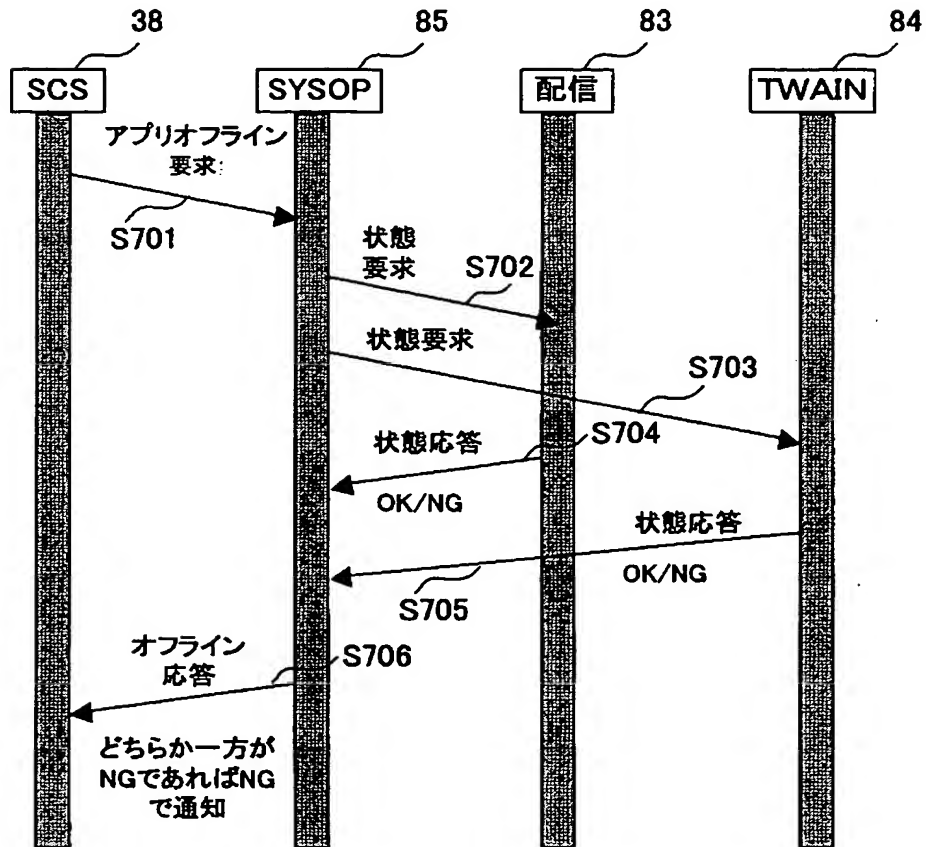
【図 20】

## スキャナアプリの内部構造の概略図



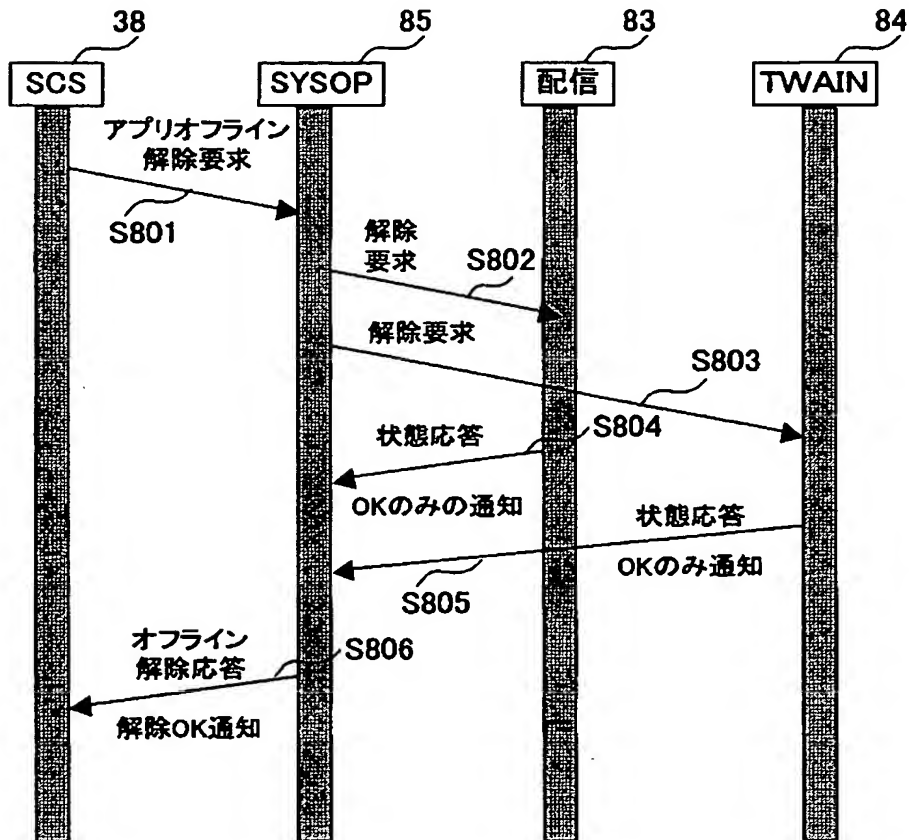
【図 21】

## スキャナアプリのオフライン処理イベントシーケンス図



【図 22】

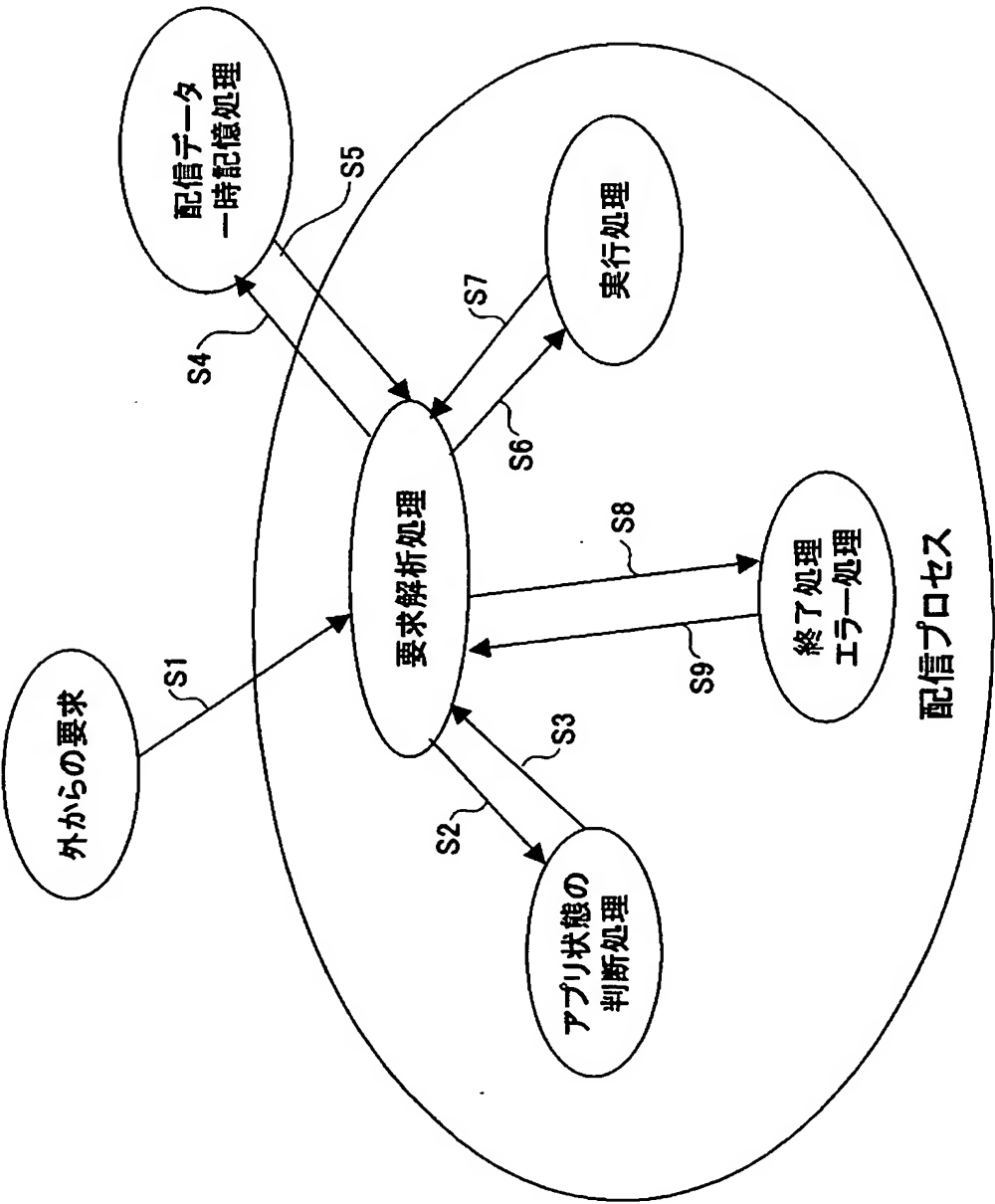
## スキャナアプリのオフライン解除イベントシーケンス図





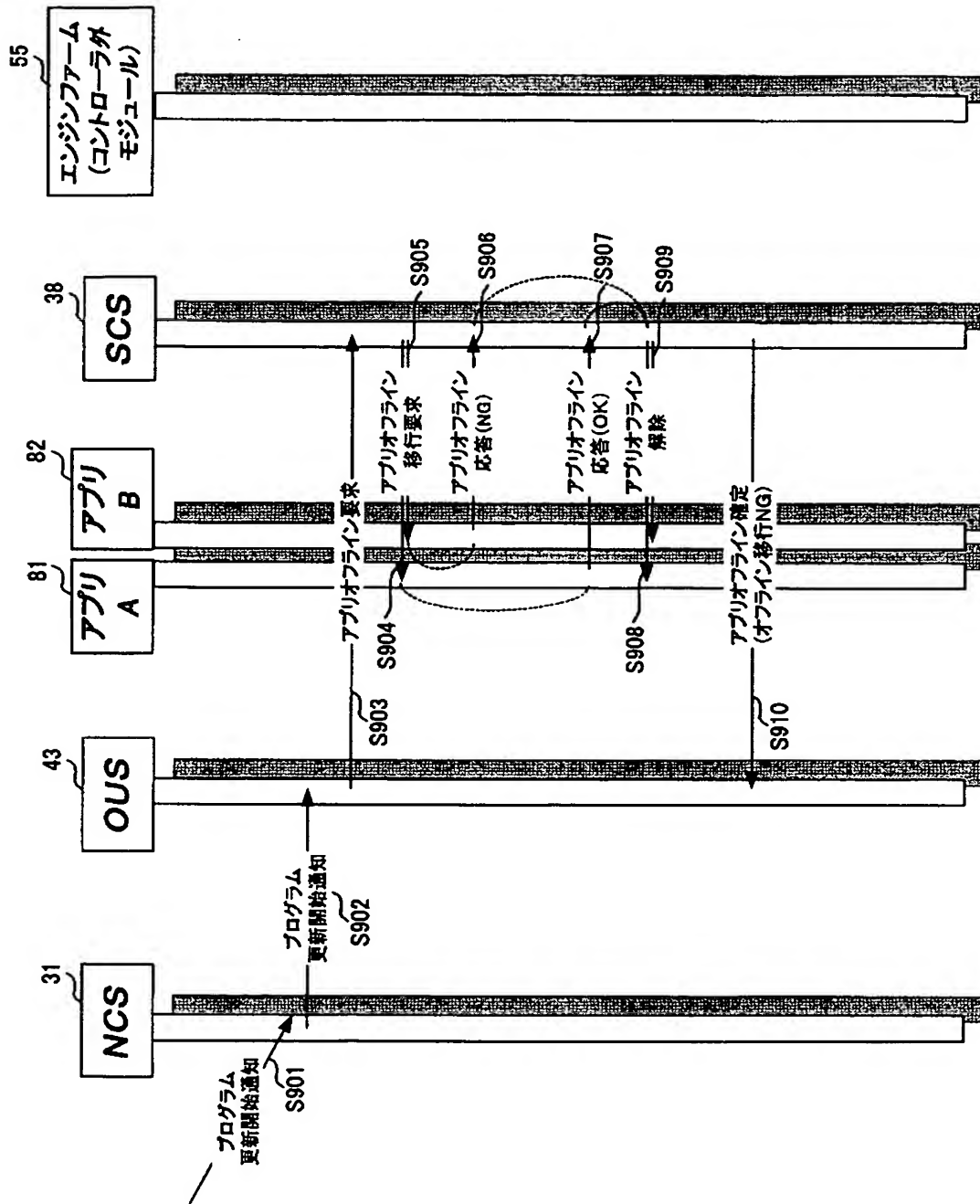
【図 23】

スキャナアプリのオフライン状態説明図



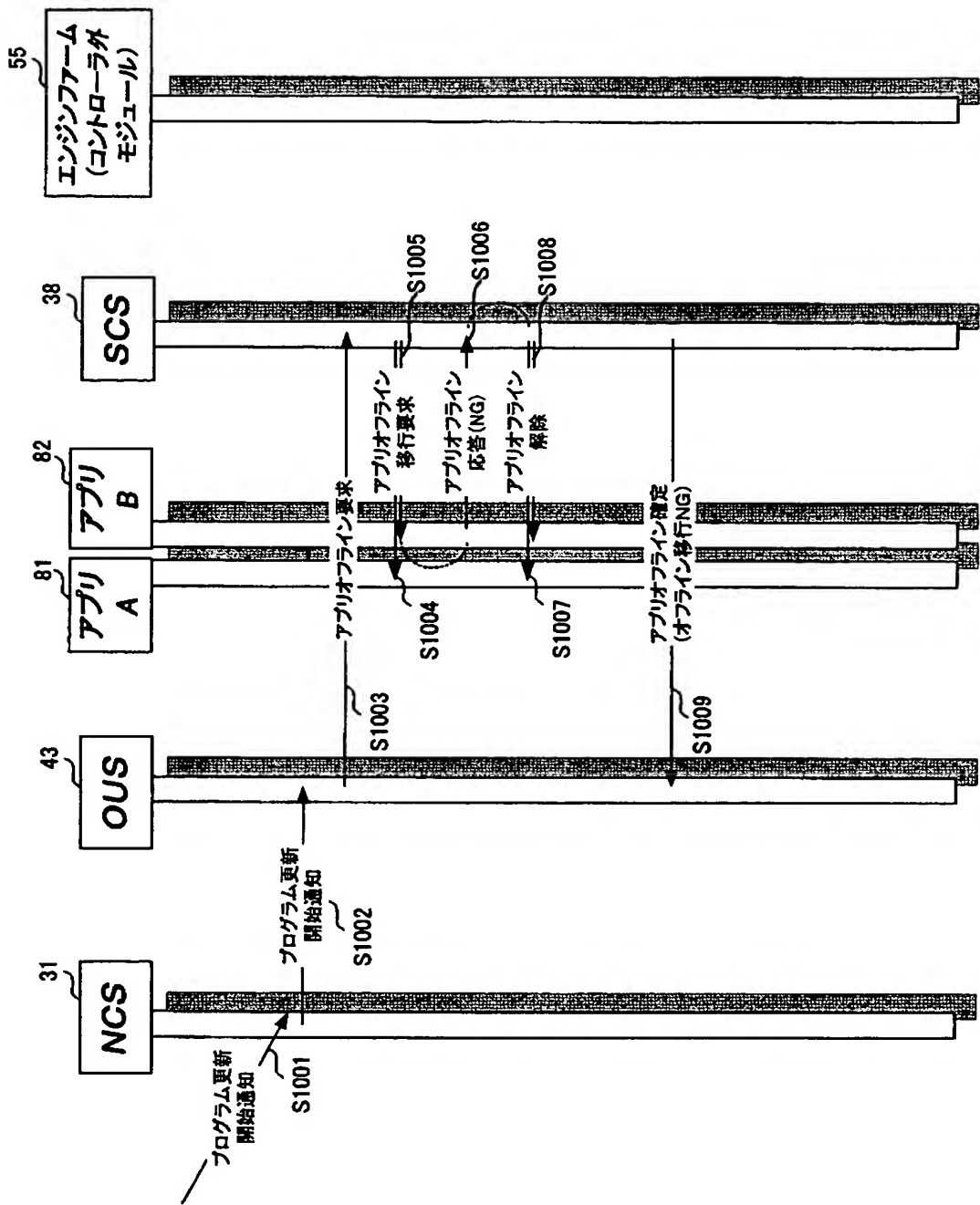
【図 24】

オフラインNG時の処理を示すシーケンス図



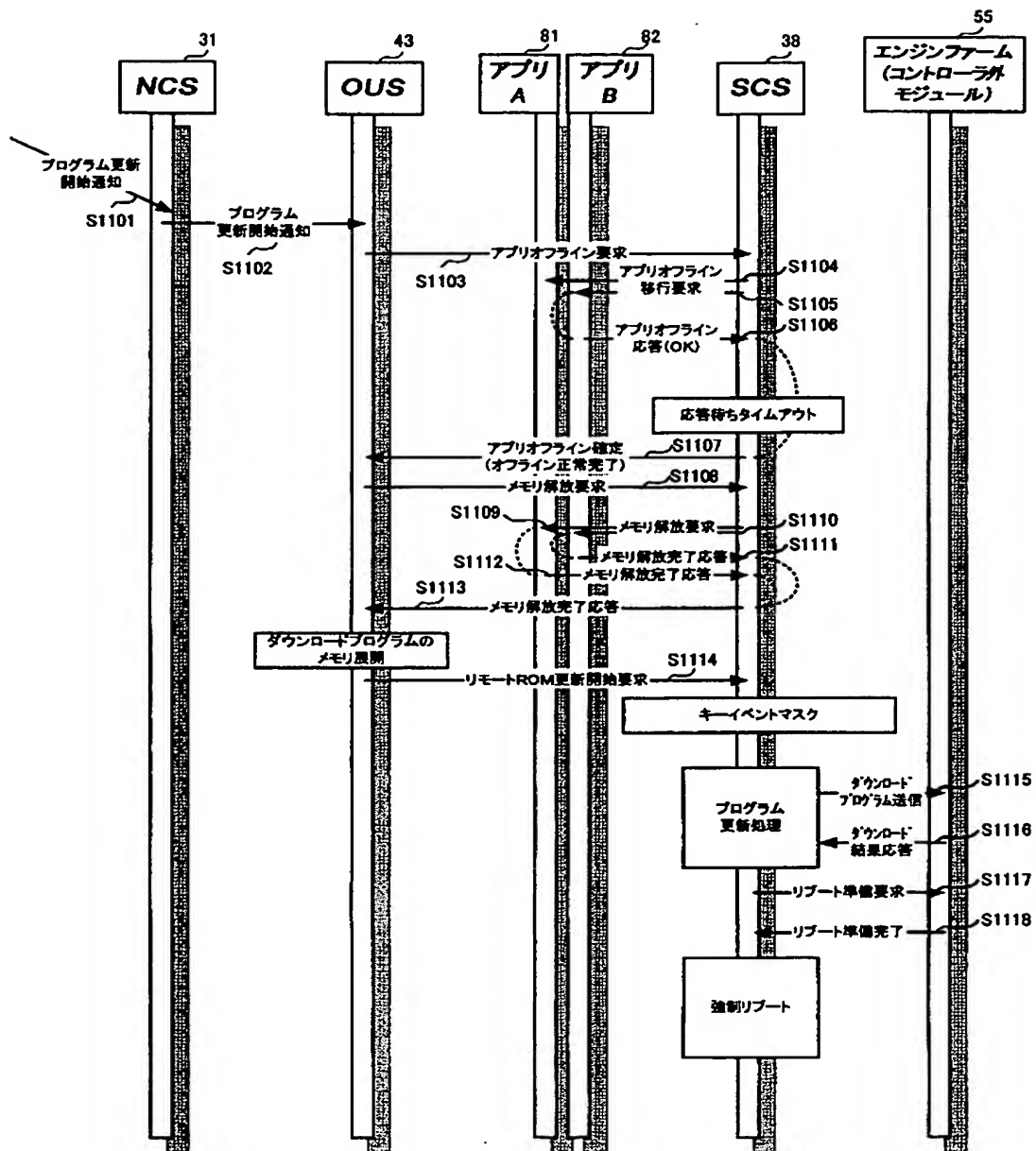
【図 25】

オフラインNG時の処理を示すシーケンス図



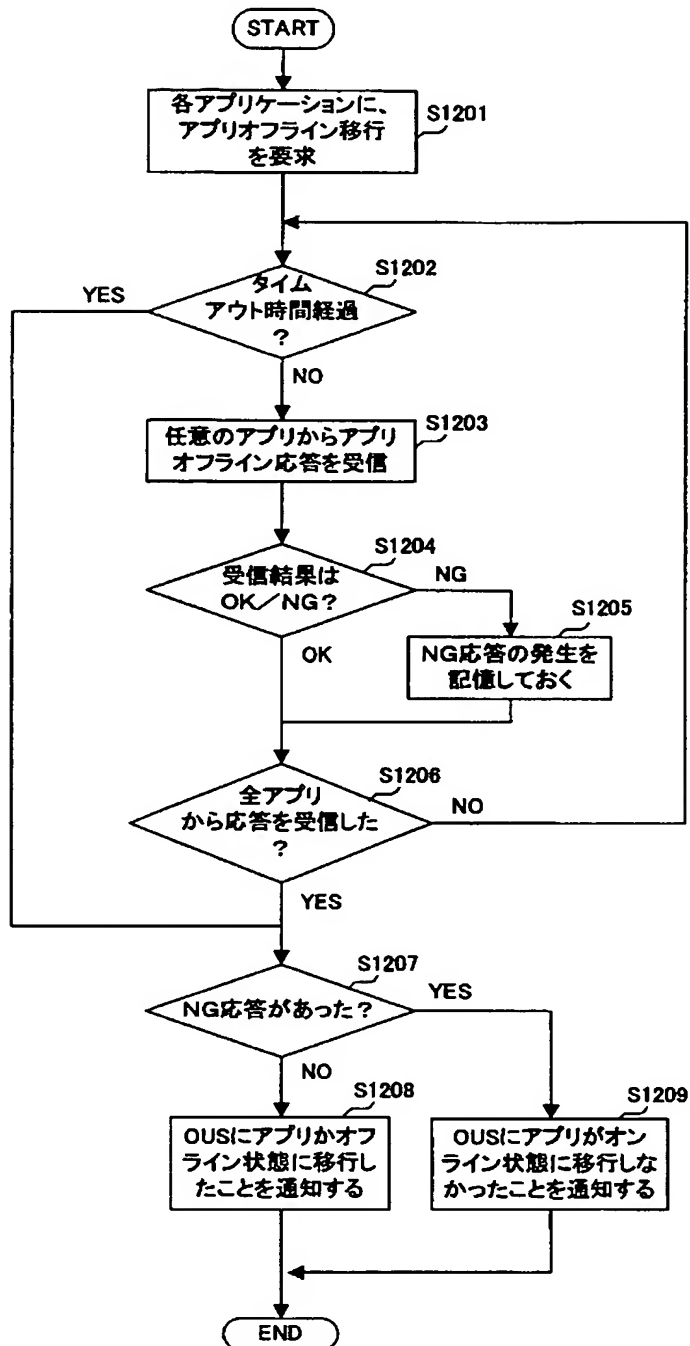
【図 26】

タイムアウト処理を示すシーケンス図



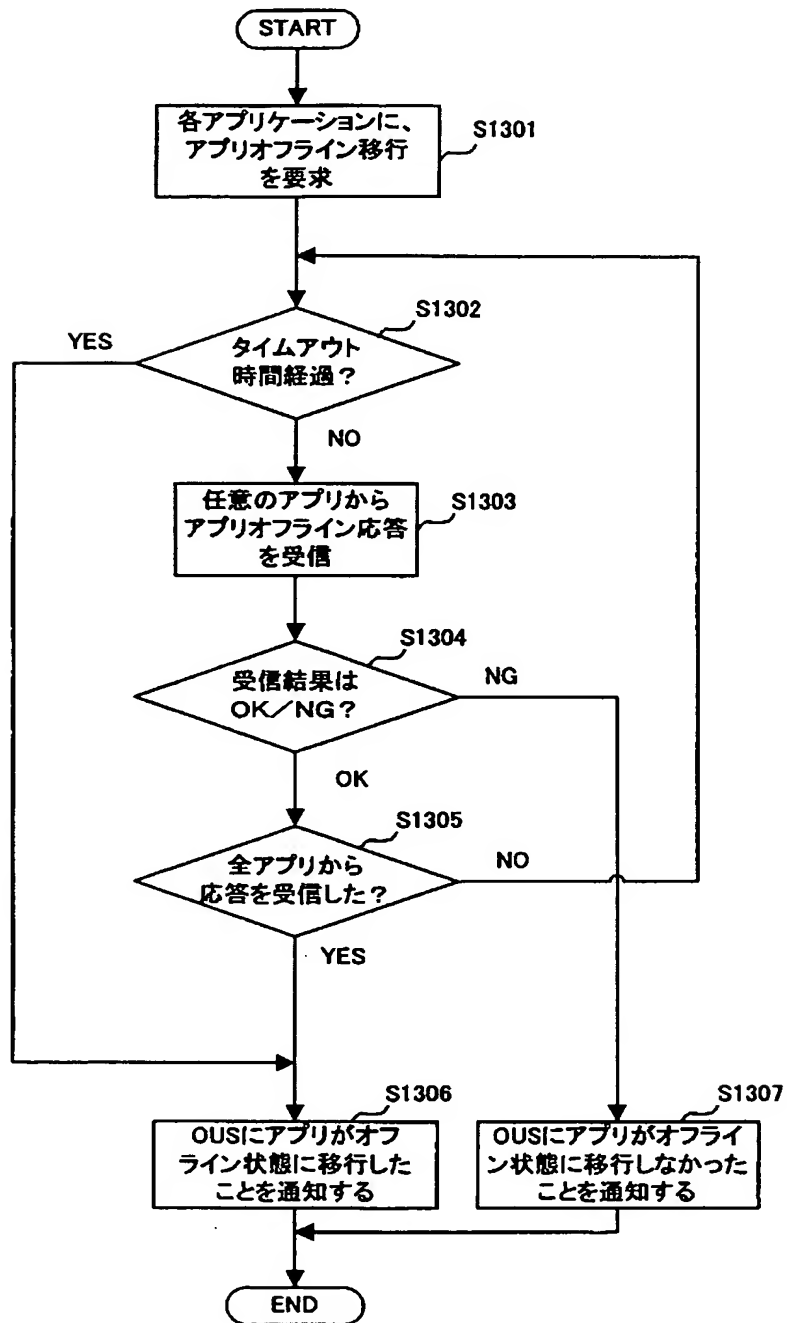
【図 27】

## SCSの処理を示すフローチャート



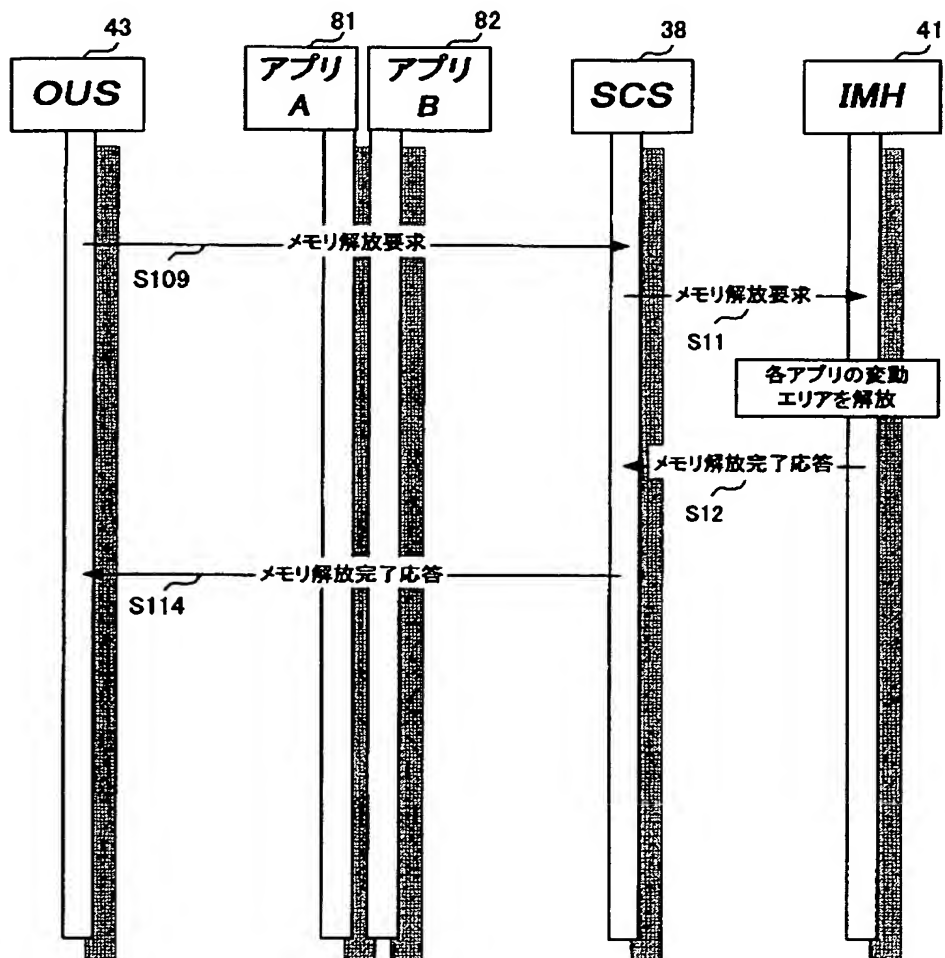
【図 28】

## SCSの処理を示すフローチャート



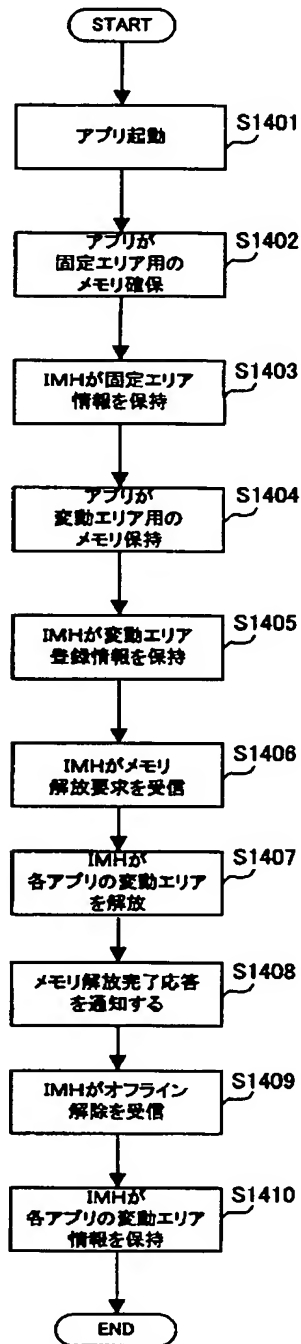
【図 29】

IMHの処理を示すシーケンス図



【図 30】

## IMHの処理を示すフローチャート





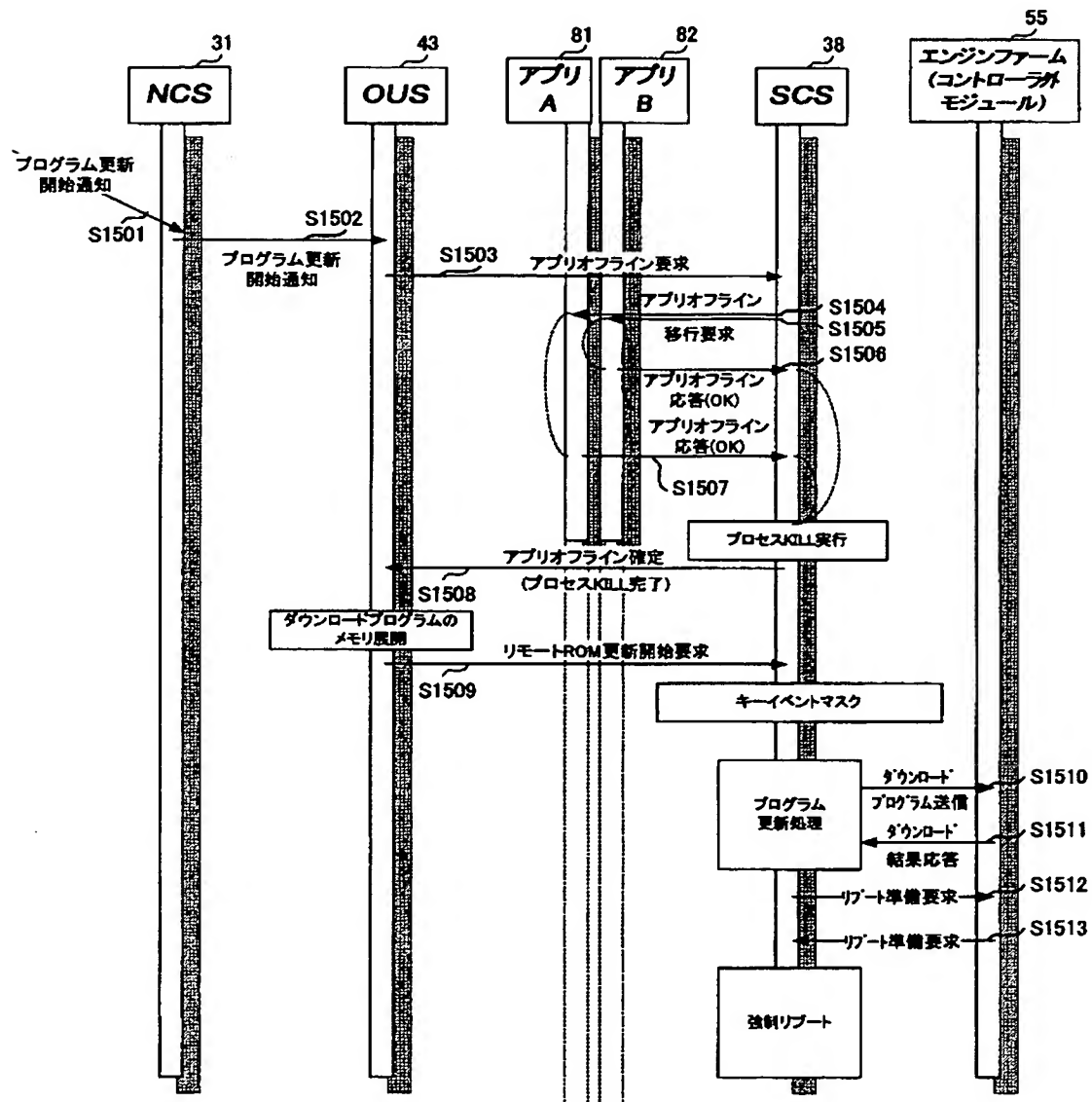
【図 3 1】

記憶領域情報を示す図

アプリケーション名	アプリA	アプリB	アプリN
固定エリア先頭アドレス	0x100000	0x250000	0x650000
固定エリア末尾アドレス	0x1FFFFFF	0x3FFFFFF	0x7FFFFFF
変動エリア先頭アドレス	0x200000	0x400000	0x800000
変動エリア末尾アドレス	0x24FFFFFF	0x47FFFFFF	0x80FFFFFF
...			

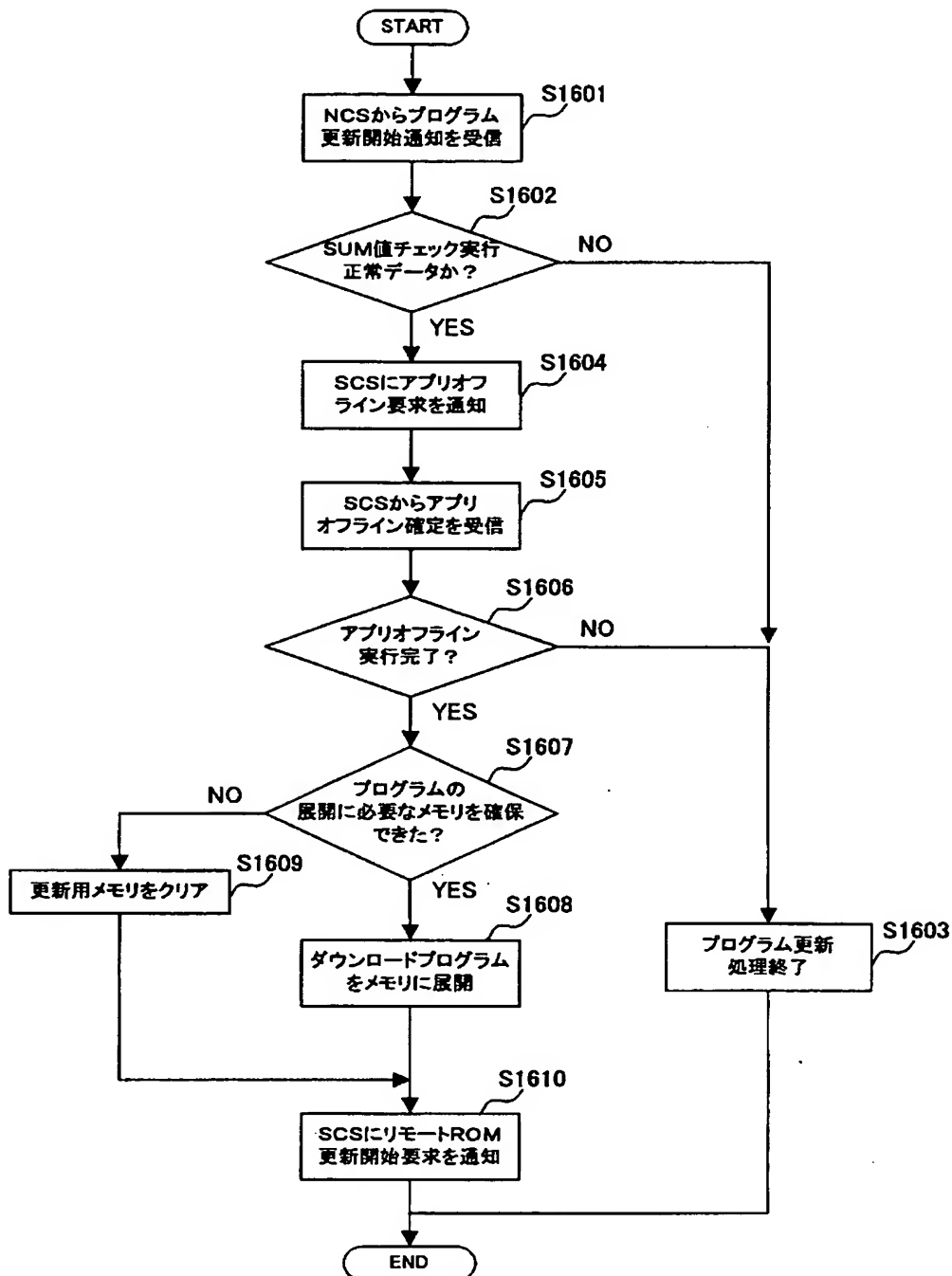
【図 32】

プログラムを更新する処理を示すシーケンス図



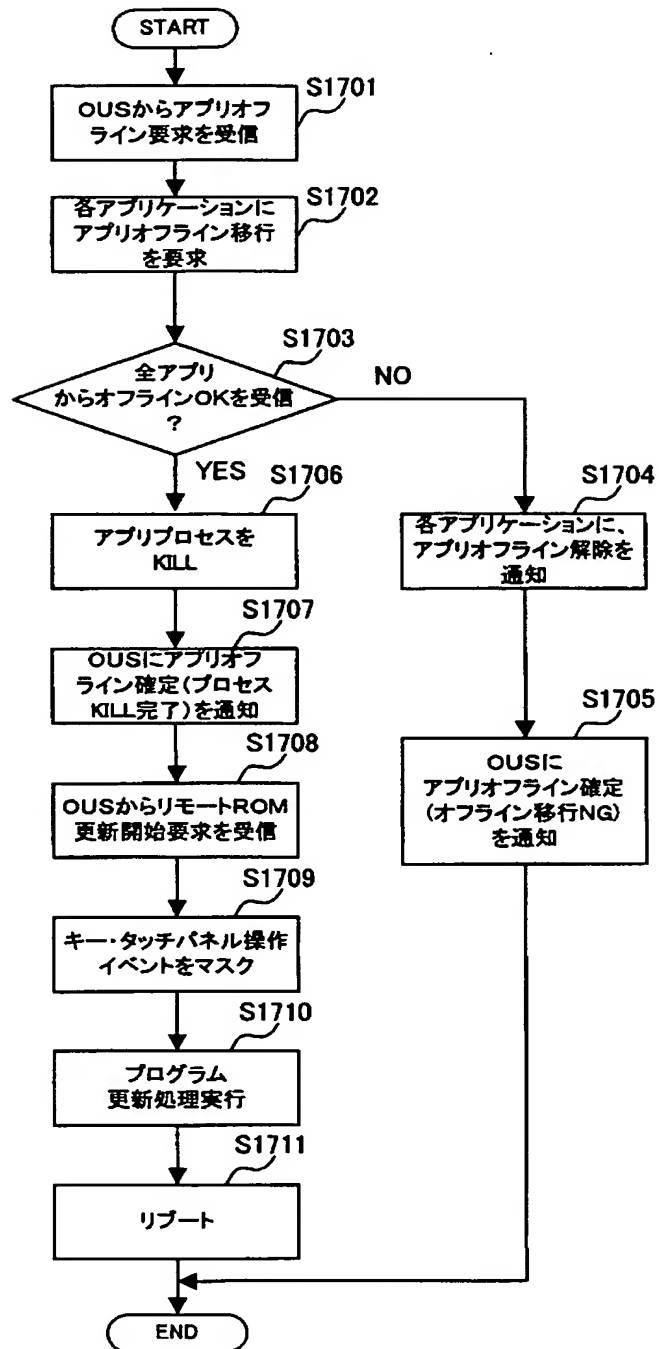
【図 33】

## OUSの処理を示すフローチャート



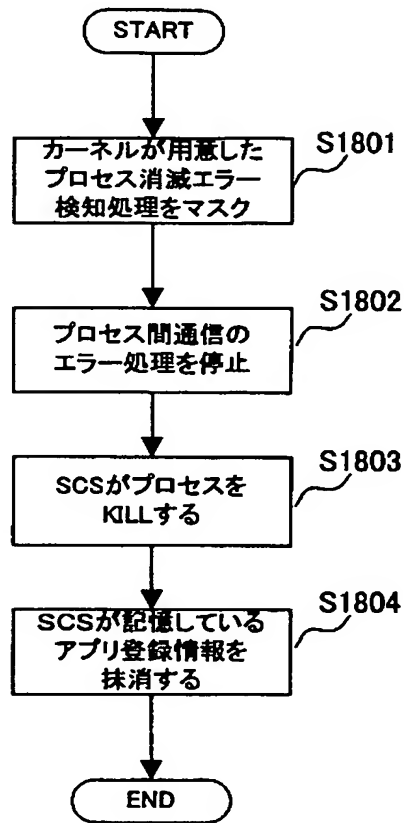
【図 34】

## SCSの処理を示すフローチャート



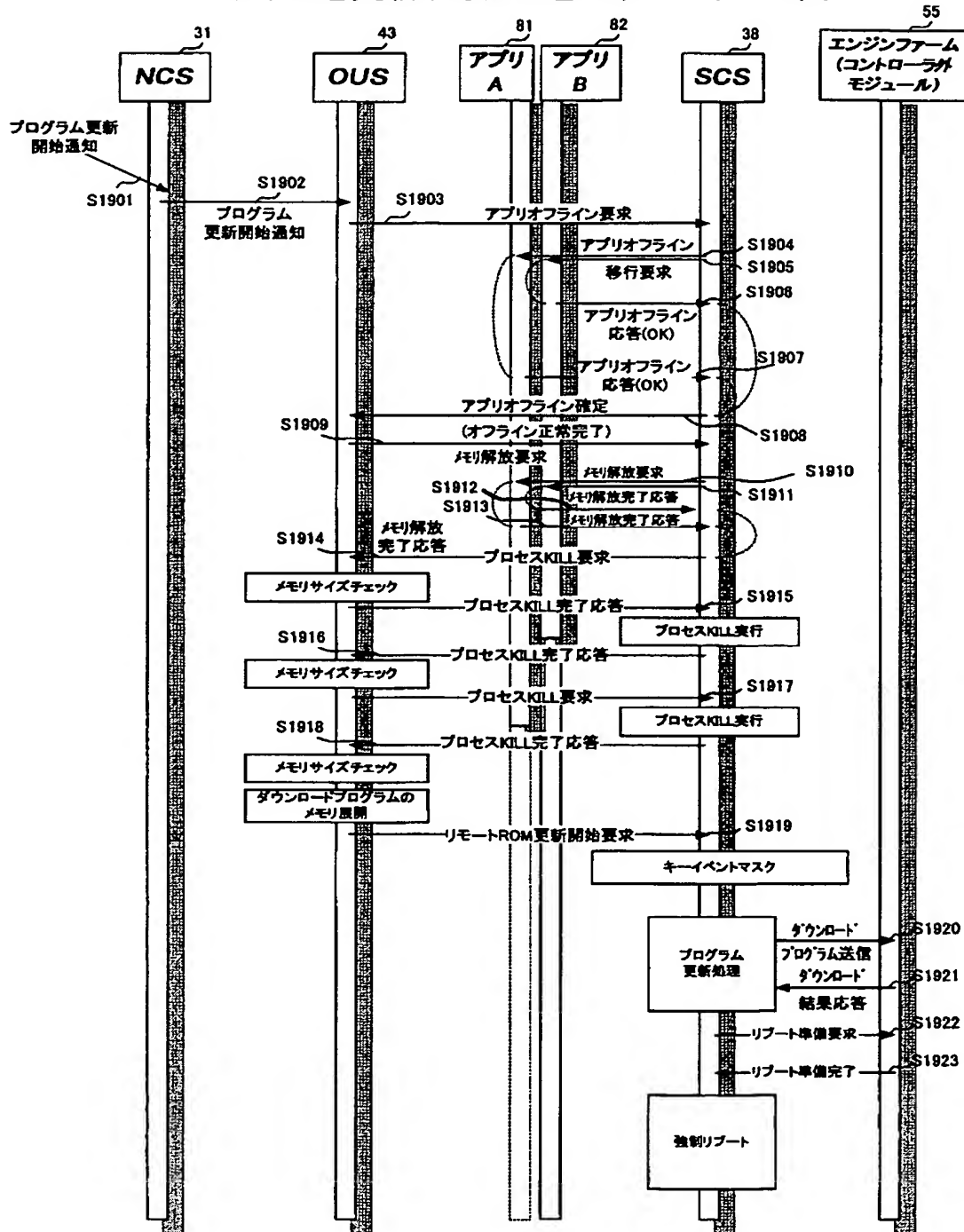
【図 35】

## プロセスの消滅の処理を示すフローチャート



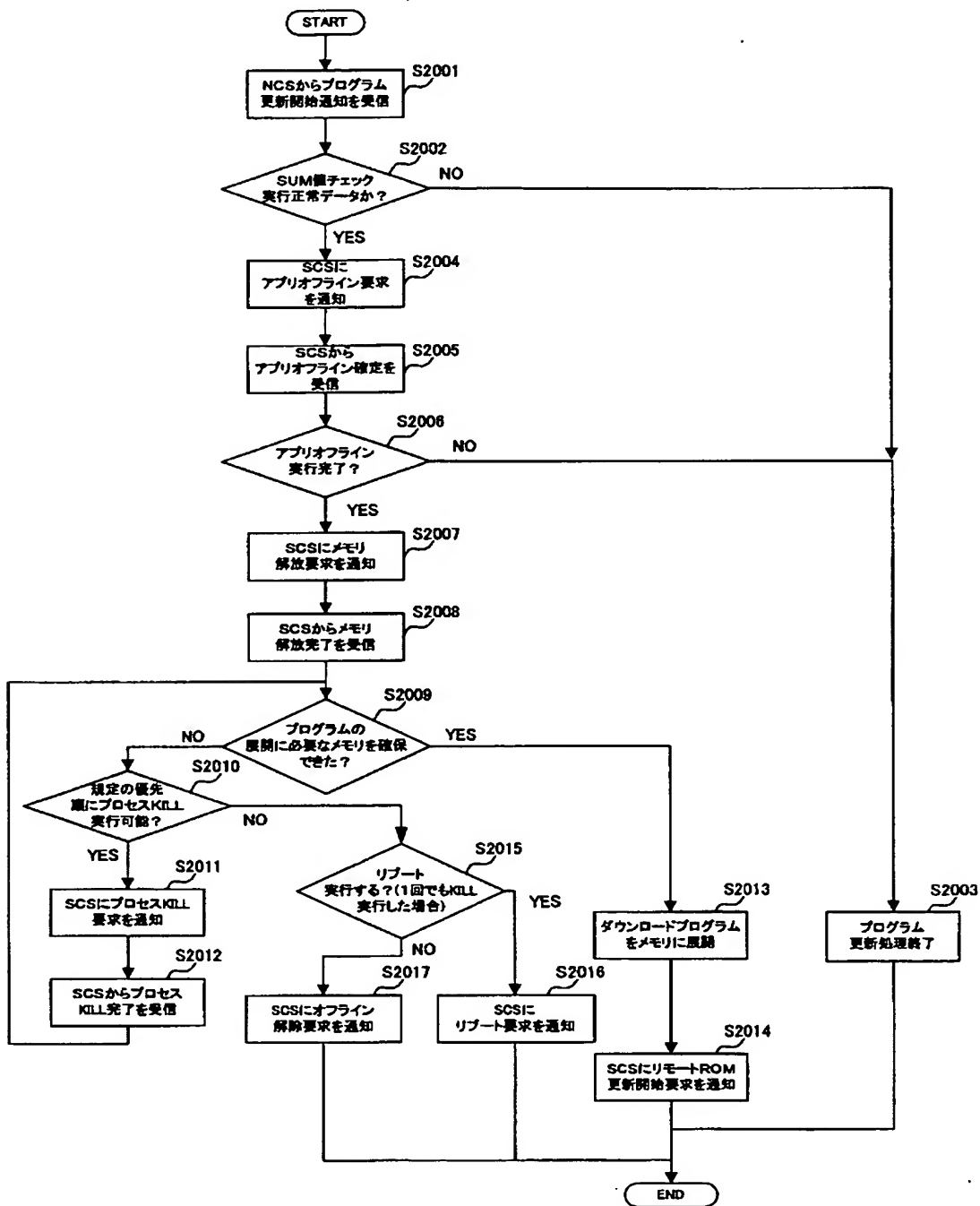
【図36】

プログラムを更新する処理を示すシーケンス図



【図 37】

## OUSの処理を示すフローチャート



【図 38】

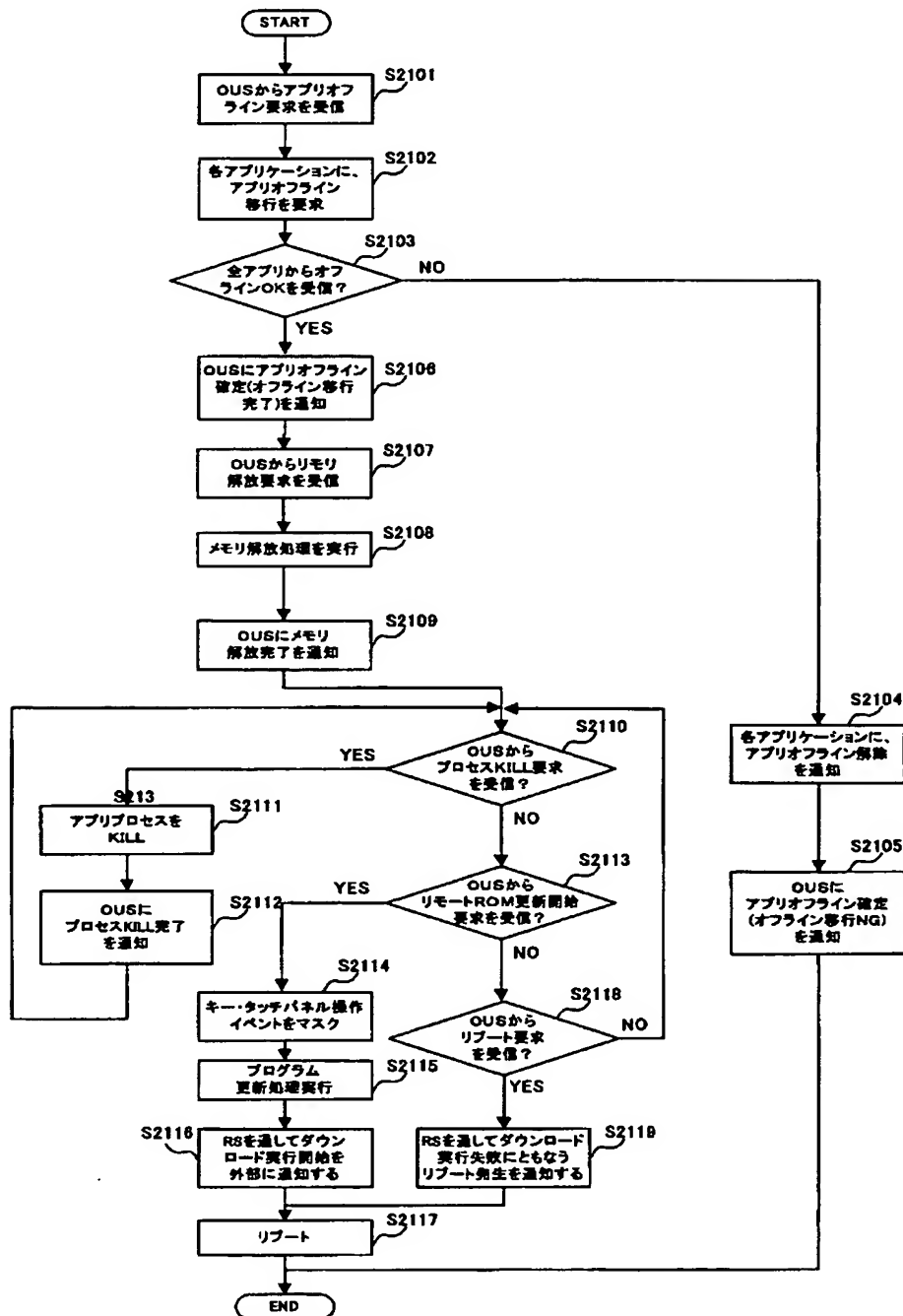
消滅順番を示す図

消滅順番	デフォルト	メモリサイズ	共有メモリ
1	ネットファイル	コピー	コピー
2	スキャナ	FAX	プリンタ
3	FAX	ネットファイル	FAX
4	コピー	プリンタ	スキャナ
5	プリンタ	スキャナ	ネットファイル



【図 39】

## SCSの処理を示すフローチャート



【書類名】 要約書

【要約】

【課題】 画像形成装置に搭載されているメモリ資源を有効的に利用する画像形成装置、記憶領域確保方法を提供する。

【解決手段】 画像形成処理で使用するハードウェア資源と、画像形成に係る処理をプログラムに基づき実行するプロセスとを有する画像形成装置において、前記プロセスの処理の実行を制限させた状態であるオフライン状態に前記プロセスを移行させるオフライン手段と、前記オフライン状態に移行した前記プロセスが使用していた記憶領域を解放する記憶領域解放手段と、前記記憶領域解放手段が解放した記憶領域を、データを展開する記憶領域として使用するデータ展開手段とを有する。

【選択図】 図 8

特願 2 0 0 3 - 2 9 1 0 4 5

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 6 7 4 7 ]

1. 変更年月日

1 9 9 0 年 8 月 2 4 日

[変更理由]

新規登録

住 所

東京都大田区中馬込 1 丁目 3 番 6 号

氏 名

株式会社リコー

2. 変更年月日

2 0 0 2 年 5 月 1 7 日

[変更理由]

住所変更

住 所

東京都大田区中馬込 1 丁目 3 番 6 号

氏 名

株式会社リコー